

Marjo Maula: Master Thesis

Software Delivery Concept

Helsinki Metropolia University of Applied Sciences

Business Informatics

Date 25.05.2014

Author(s)	Marjo Maula
Title	Software Delivery Concept
Number of Pages	103 pages
Date	25 May 2014
Degree	Master of Engineering
Degree Programme	Master's Degree Programme in Business Informatics
Instructor(s)	Hannu Vähäsaari , Deputy CEO in Profit Consulting Oy Thomas Rohweder, Principal Lecturer
<p>The purpose of this study was to create a software delivery concept for ERP in IT-project within Profit Consulting Oy. Profit Consulting is a small size consulting company of 50 employees providing services in Project Management, Architecture, Integration Services and Software Development. Software Delivery Concept was created because in startup IT-projects concept for delivering the software is missing.</p> <p>Both qualitative and quantitative research methodology was utilized in this study. The qualitative research data consists of six in-depth interviews. The interviews were done with project people and people who are working in global projects. Quantitative data research data was gathered with a questionnaire. Six questionnaires were sent for IT-project professionals in international companies, four responses were received.</p> <p>The results from both the research indicate similar type of challenges faced in the projects: requirements are unclear, scope creep happens, communication is insufficient, full traceability is not visible and test automation is not fully implemented. Test automation is seen as a key opportunity to speed up release cycle times and manual testing is useful in exploratory and e2e (end-to-end) testing.</p> <p>The author recommends that a couple of focus areas in scope will be improved in the next release, for example communication, scheduling and measuring, traceability from requirements to deployment and automation in testing and in configuration.</p>	
Keywords	Software, Delivery Concept, Software Delivery Concept

Contents

Abstract

1	Introduction	4
1.1	Background	4
1.2	Business Problem and Target of thesis	9
1.3	Scope	11
2	Research Approach	12
2.1	Research Process	12
2.2	Data Collection and Analysis	14
3	Current State Analysis - Project Framework	16
3.1	Current Project Framework - Process	16
3.2	SWOT analysis, Project Framework	17
3.3	Current Development and Release Process - Back-end system	20
3.4	SWOT analysis - Current Release Process (Back-End)	23
4	Project Basics and Best Practices	27
4.1	IT Project Facts	27
4.1.1	Why ICT projects fail?	28
4.1.2	Project measures and metrics troubles J. Kendrick (PMP, CMC)	29
4.1.3	CIO - Pearl Zhu - Ten Root Causes of IT Projects Fail	29
4.1.4	Why IT projects fails - IT Business Edge?	30
4.1.5	IT Project Success Factors	32
4.2	Project Management	34
4.2.1	Project Management: Foundation and Planning by Karl Wieggers	34
4.2.2	Transforming to Agile Project Management	37
4.2.3	Project Manager's Agile Survival guide by Rally Software	37
4.2.4	Mapping of Waterfall Project Management to Agile Practices	38
4.2.5	"Subway map" by Agile Alliance	39
4.2.6	Kanban: by David J. Anderson	41
4.2.7	Five challenges to Agile Planning	45
4.3	Software Testing	46
4.3.1	Pre-conditions for software testing - Entry Criteria	46
4.3.2	Software testing in general	46
4.3.3	Manual Testing	50
4.3.4	Regression Testing and Automation	51

4.3.5	Performance Testing	53
4.3.6	Security Testing	55
4.3.7	Software Delivery	57
4.4	Alternative Software Delivery Concepts	57
4.4.1	"WaterScrumFall" - What is that?	57
4.4.2	IBM introducing Global Delivery Excellence Best Practices	59
4.4.3	Two kind of delivery models	60
4.4.4	Continues Delivery	60
4.5	Conceptual Framework	62
5	Creating a Prototype Concept	73
5.1	Creating a Prototype Concept and Theme Interviews	73
5.1.1	Project Management	74
5.1.2	Release process	79
5.1.3	Testing	81
5.2	Delivery Concept	84
5.2.1	Project Framework	84
5.2.2	Development and Release Process - Back-end system	86
6	Testing a Prototype Concept	87
6.1	Concept Proposal: Software Delivery Concept	87
6.1.1	Proposed Improved Project Framework	87
6.2	Project Metrics and Measures to follow up	91
6.3	Improvement based on the test / Lessons Learn	96
6.3.1	Corrective action list / Project Management Practises	96
6.3.2	Improvement ideas for Release Process - Back-end system	99
7	Conclusions	101
7.1	Summary of the Project	101
7.2	Evaluations of outcome against the target/object	104
	References	106
	Appendixes	110
	Appendix 1: Research Process (2 Research Approach)	
	Appendix 2: Data Collection and Analysis	
	Appendix 3: PMBOK Practices mapped to Agile Practices by Rallydev.com	
	Appendix 4: Mapping Integration Management to Iterative Process	
	Appendix 5: Two Software Delivery Models	
	Appendix 6: Project, Release and Testing questions	

Appendix 7 - 12. Theme Interviews

Appendix 13: Questionnaire for project work

1 Introduction

1.1 Background

Background for the Master Thesis comes from the project in which the new Insurance Company was created. The biggest business driver in the strategy is to do the purchasing in web.

In some smaller size IT- service providing companies or in start-up customer companies the concept for Software Delivery process is missing. The package and guidelines for delivering the Software Release rapidly with low costs, less time with minimum resources is missing.

Like in many other start-up companies software delivery concept and common practices for project management to be created and that was done also in this project.

Information and Communication Technology (ICT) changes

Over the past 10+ years the opportunity and the demand has increased tremendously for High Tech industry e.g. Mobile industry, Social Media, Gaming, IT Consulting and Health care.

Some reasons why ICT changes has happened are globalization, new innovations, current advanced Technology trend, automation, enhanced ROI - Reduction of Cost of production, manpower and pressure to increase and maximize the profits (internal efficiency) in short and long term.

Technological Factors impact for IT-Technology availability and usage and ICT sector future trends. Technological Factors can affect costs and quality, lead to innovations and give the impulses for outsourcing decisions and strategies in both worldwide and locally. Technological changes generate new opportunities in service sector for service providers and industrial companies. They can optimize processes by implementing new technologies to change the existing IT systems or replaces and substitutes old systems. Level of System (IT) Automation and Web technologies give not only the new opportunities but also give the needs for new services or give an impulse to improve existing services. Social Networking Technologies provide a communication channel for the better customer service, advertising and networking.

Governments influence for education and public and private services. Many governments look for IT savings. This is consequence of economical situation and willingness to harmonize of governmental and communal IT infrastructure and systems. Transparency is needed in healthcare and in public services and in private sector in all countries.

Snapshot of ICT Business in Finland

During the 1990s the economy oriented heavily towards ICT (information and communication technologies), and by the end of the decade Finland was the most ICT specialized economy in the world. From 1990s to 2013 ICT have had very important role in business and economics and the growth will continue. Finland is one of the leading knowledge based economies, not only one of the most open economies (Ylä-Anttila, 2012).

15 Feb 2012 UNCTAD said Finland leads in ICT jobs. Nearly 10 per cent of Finland's population worked in information and communications technology (ICT), making Finland the world leader in ICT employment. The agency said ICT is increasingly playing a role in developing economies.

Six top tech trends in 2014 by CIO Insight

Digital Technology and always-on connectivity have created new and impressive opportunities for the enterprise. IT environment is becoming more complex because of the requirements for the accessibility (anywhere and anytime). IT requires new and radically different approach and environment. The focus is in highly targeted functionality through apps delivered via enterprise app store.

"The explosion of mobile devices has increased the number of tech-savvy employees over the past five years, all of whom are pushing to consumerize the way that IT departments operate", observes James Gordon, vice president of information technology at Needham Bank in Massachusetts. "Employees want to be able to download the apps they need, and they don't want to have to ask for download permissions or access rights in order to get their job done."

Above requirement translates to the need to monitor the network and how apps and data are used in there and reduce security risks.

All this requires business and IT leaders to "fundamentally rethink how they deliver applications and services", says Tiffani Bova, a Gartner vice president and distinguished analyst.

Garner predicts that in 2014 the personal cloud will replace the PC at the center of users' digital lives. "Personal clouds offer a much more flexible and productive way to manage applications and data", says Bova, (Greengard, 2014).

Improving Business - IT Alignment by Risa Fogel (Senior Managing Director at Cushman & Wakefield)

We wouldn't talk about business and IT separately. Is it a failure of communication? Additional for relationship in success talented people should provide a transparency into IT activities and spending to avoid the gap between IT and business and help on conversations (Fogel, 2011).

Risa Fogel advises to have three different topics to make the understanding better and communication easier. The first topic is about changing the vocabulary. Instead of talking technology and using technical terms conversations should be tied to the business goals and use a business frame of reference, how to solve the existing problem and enable future growth.

The second topic is about establishing business-facing roles, business relationship managers. The business relationship managers encourage for cross-functional collaboration. They also balance expectations, priorities and service levels. The role is essential to explain business and technology challenges with equal clarity.

Establishing the governance processes is the third topic. It will provide the visibility into decision making and spending. The transparency and visibility leads to stronger relationships and partnership and helps on identifying the common goals. All this improves the communication and knowledge about IT challenges to meet the requirements and demand. Providing a view to IT doings helps on decision making. Two-way communication will be achieved by adopting these strategies.

Trends Shaping Product Delivery by Eric Winquist (Jama Software)

Eric Winqvist is CEO at Jama Software. He presents the leading ideas of what type of challenges teams can face and he reveals strategies how to improve the product delivery in the organizations.

The fact in product delivery is that 70 % of the organizations release their products quarterly or more often and of that 70 % monthly or more often than that releases 58 % of the organizations.

Requirements are the number one reason for the delays. Product delivery delays are also caused by slow decisions. When there is a question of processes and what type of processes are used for the teams, almost 20 % are using different processes for the different teams. This can cause confusion, misunderstandings, delays in communication and actions in the projects when the steps are followed a bit different ways.

For the customers most important things are the products what are delivering the core business value and the important, number one indicator to measure the progress is improved customer satisfaction. Aligning the delivery processes and finding the core business value, how to do that? Finding the core business value can be made by identifying the steel thread, determining what and how the success look like and by evaluating against new information.

Companies should define WHY. This simply means answering the following questions:
Why are we building this product? Why are we building this feature?
Why did we make this decision? What are the outcomes?

When there is a need for the change rethink it. There are some things what can help on rethinking. Rethink the changes by providing the context, by empowering the decision making, by linking people to the work and by understanding why, (Winqvist 2011).

There are two additional very important things to add for the five tips given by Eric Winqvist (Jama Software) and those are stop reinventing - start reusing and review early and often.

Modern Product Delivery by Eric Winqvist (Jama Software)

In modern product delivery the focus is on the customer. The target is to find the ways to be faster and managing the complexity and still be innovating and better understand the customer needs.

Define the why. In addition to defining the features and functions it is better to let the customers understand why they need the functionality and what problems is solved by implementing them, what are the achievements. When the product team understands why and what they will do the better decisions are done to fulfill the gaps.

When the focus is on the core business value and technical team has adapted the demand they will make the right decisions through the technical complexities to obtain the business outcomes, what best outcomes the product should deliver.

Sharing the information and reusing it technical team become faster and product complexity is avoided. The code reuse happens first. The customers are using design artifacts, specifications, test cases, data sheets and process information.

Cycle times can be reduced by reusing the information from libraries and launch by launch updating the new information and improving the cycle.

Reviewing early and often with the stakeholders improves the speed and quality. The key is to keep the process iterative and small. Today there is a never-ending flow into the product team. At fist are needed modern tools to support sharing the work in real time. The traceability through the project or delivery is essential.

In September 2013 Jama Software has commissioned Forrester Consulting to do the examination about organizations delivering the products, exploring the challenges faced in meeting the needs of customer in rapidly changing competitive environment. Forrester developed a hypothesis and delivered five key findings.

The first very important key finding is that the product delivery has become in a critical advantage to be a successful and competitive market leader. Product delivery garners lots of attention; it has been highlighted into a strategic role in many organizations. Product delivery is a cross-functional activity and it spans all the important areas like executive management, sales, marketing, services, support and operations as well as all the product management related processes (development, IT, QA and release management functions).

The crucial thing to recognize and understand is what the customer needs are. When requirements are unclear and changing continuously it will cause troubles for the product delivery. If there is not a clear understanding what needs to be build and if the feedback is not got early enough the end result in product delivery will be unsatisfied. There will be delays in delivery and time, effort and money will be wasted in a release or a project.

The priorities should be agreed clearly by stakeholders. The objectives, assumptions and possible solutions should be clarified otherwise it will lead to a lack of focus. If the goals are not aligned the product delivery teams are not able to resolve the conflicts.

An effective collaboration is must in cross-functional organization which spans the roles, teams and geographies. Successful product delivery requires wide expertise and experts in right places. Modern products are complex and teams are globally distributed for different regions.

Delays in decisions reduce speed of the delivery and impair negatively for quality.

Companies having a rapid product delivery are competitive differentiator. Communication obstacles to be removed to speed up the deliveries and help companies to meet customer needs, (Winqvist 2014).

1.2 Business Problem and Target of thesis

Business Problem:

In some smaller size IT- service providing companies or in start-up customer companies the concept for Software Delivery process is missing. The package and guidelines for delivering the Software Release rapidly with low costs, less time with minimum resources is missing.

Some of the companies either do not have or do not use the software delivery concepts for the similar type of IT-delivery projects or define the process practices every time based on the experience from the previous projects. In some companies projects are following the best practices what responsible Project Manager or Agile teams have used based on their previous experience and from the previous projects. If many processes and much type of practices are used in different areas end to end process management can become challenging. That makes also longer term (4 to 6 months) project portfolio planning and management and resource planning more challenging.

In start-up companies concept to be created and can be implemented based on the best practices from the project management and from the Agile practices.

In the example project case there are three different suppliers and three different types of way to work and release the software to production. Some are following mostly or partly agile methods and some more traditional software delivery project. The challenge is to get these parties and development work synchronized so that the outcome gives the best possible value for the customers. Bigger the project is and more different parties involved for the project more different process and practices are also in parallel. The jungle is just in there.

It is obvious that any commercial delivery concept is quite impossible to create because there are not exactly identical projects. Benefit is to recognize those parts in software delivery projects what can follow the same methodologies and can be measured in a same way in one company.

Objective is in the Master Thesis is to create a Software Delivery Concept for the back-end system (ERP), comparable metrics for the project management and make a improvement list of the future actions.

The purpose is to create the concept proposal (Prototype) based on the current state analysis (SWOT: Project Management and SWOT: Release process), Theme interviews and information gathered from different sources and from literature.

Software Delivery Concept proposal is piloted. Piloting and testing is done for the prototype solution. Got experience and outcome is analyzed. Outcome from the pilot is adjusted for improved Software Delivery Concept (Project Management Process and Release Process) Project Metrics and Measure examples are presented to give an example how internal project follow up can be done. Improvement ideas and lessons learned from the project are collected the corrective action list / improvement ideas are provided. Focus in the project is in project management process and in release process.

1.3 Scope

The scope of Software Delivery Concept in this Master Thesis is covering an IT-project which IT-architecture consists of the following systems and integrations:

- Core System - **Back End system (ERP)** including different modules for business use including Service Center (for example: Quotations, Sales, Accounting and Claims)
- **Partner channel system** what is available for different business partners, at first place for car dealers
- **WEB-front end** for consumers (individual customers)
- **Integrations** between these three systems (Back End, Partner Channel and Front End (WEB)) and
- **Integrations for external systems** (for example banks, authorities, bookkeeping system, credit check).

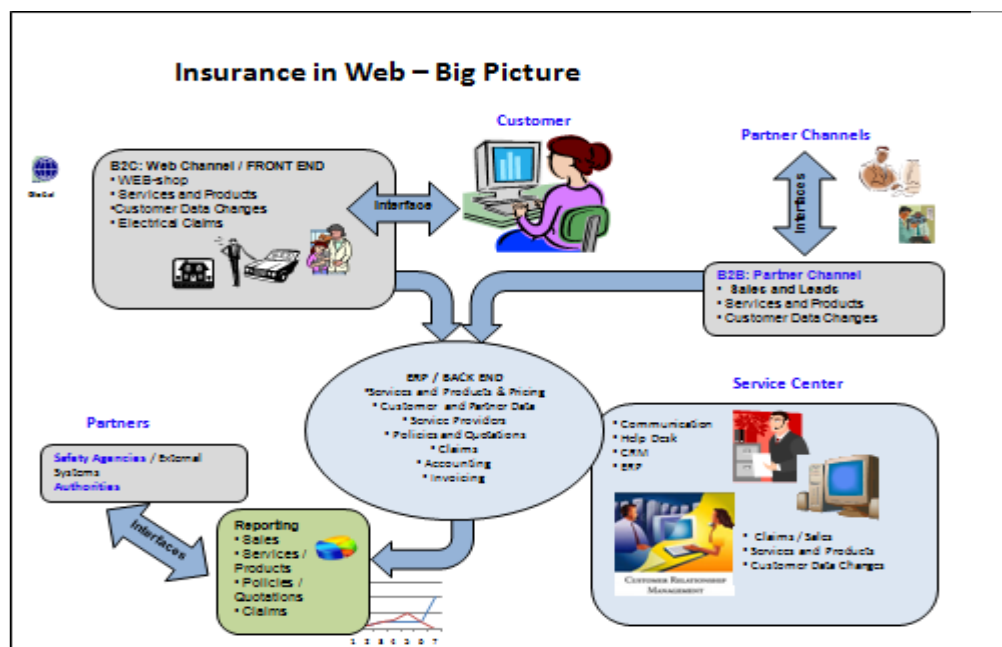


Figure 01. Scope: Insurance in Web - Big Picture

In scope there are multiple suppliers what are delivering “own” parts for the project scope. One system supplier is international and two are the local ones. Project languages at the first phase are English (communication with international supplier) and Finnish (communication language within two other suppliers). The customer is from Finland and consumers are mostly Finns. Core Software itself is translated / localized for the local language but also English version is tested. In scope there are several integrations for outside systems and integration layer between Core Back End System (ERP) and WEB Front End system.

In the example project there are three different suppliers and three different type of way to work. Some are following mostly or partly of agile methodologies, some more traditional software delivery project approach.

Current process or some parts of it and current practice to be aligned because the process is not providing sufficient visibility and information for the company management and for the project team.

The critical facts and factors of IT project delivery and project complexity give the input, needs and requirements for developing the Software Delivery Concept.

Exclusions

Out of scope are excluded infrastructure, technical architecture and environment management, detailed level technical information and documentation, security, excluded external systems and cost management.

2 Research Approach

2.1 Research Process

The research process is following common approach what is used for research, step by step process from problem identification, through current state analysis and comparable practices creating the concept and pilot and repeating if/when necessary. Research process - Appendix 1.

The topic for the Master Thesis: Software Delivery Concept; was selected because of the own interest, some knowledge learned and some challenges faced during the company start up project. After researching the different topics about the project management, practices and different phases in one project and by interviewing the key people in the same project the pilot concept proposal was created to improve the current practice what are in place in this specific case.

Problem Identification / Target Setting

The package and guidelines for delivering the Software Release rapidly with low costs, less time with minimum resources is missing in some smaller size IT- service providing companies or in start-up customer companies. The same was faced in the project where start-up company was build from the scratch and the whole software delivery

concept was created by utilizing the known practices from the Agile and more traditional project management ways.

Objective is to create a Software Delivery Concept for the back-end system (ERP), provide an example, comparable metrics for the project management and make an improvement idea list about the possible future actions.

The purpose is to create the concept proposal (Prototype) based on the current state analysis (SWOT: Project Management and SWOT: Release process), Theme interviews and information gathered from different sources and from literature.

Research Approach

The research process is following mainly the quite common approach what is used for research, step by step process from problem identification, through current state analysis and comparable practices creating the concept and pilot and repeating if/when necessary.

Current State Analysis/ SWOT

The current Development and Release Process of back-end system, ERP is analyzed by using SWOT-analysis and descriptions for each of the identified items in SWOT-analysis are described in a deeper level. The current Release management process of managing software releases from development stage to software release in production is described in this part.

Three different phases from requirements gathering through testing and deployment in production are analyzed. The software lifecycle software release ends into the production environment where the real production end-users start to use the product.

Creating a Prototype Concept and Pilot (Theme Interviews)

Based on the results from SWOT (Strengths, Weaknesses, Opportunities and Threats) from both of the areas (project management framework and the release process (ERP)), from the facts from the conceptual framework and from the theme interviews Prototype Concept is created, piloted and tested.

In the interview circle lessons learned are collected systematically from the people who were interviewed. Corrective action list / improvement idea list is created and Project metrics and measures for the internal usage is implemented.

Theme Interview and Theme Topics have been discussed with two of people who have been in the project team in development and implementation and four people outside of the project and customer organisation but working in international IT projects.

Examine Results / Verification and Final Proposal / Conclusions

Results from the current state analysis's, literature and web references reading observation and theme interview result examination have transact sources for concept proposal creation.

From the theme interviews have got feedback what went well and what could be done better in the projects. Output from interviews have given ideas how to improve the things in the future and what type of Project Metrics and Measures for internal usage and release/project follow up.

Repeat when necessary

The process is repeatable. It can be repeated if/when necessary to improve the latest results. Deming cycle (Plan, Do, Check, Act) works; no matter in which area we are repeating the process to improve what is implemented. At the first place planning phase consists of establishing the objectives and defining methods to reach the goals, secondly what is planned will be implemented, then is measured and compared the obtained results against expected results and then improvements are done. Similar way this project is repeatable or new release process is repeatable. The scope of project or release can vary from full to partial, from large to small, resources and schedule can be different and but an idea of improving the current / latest process version with Deming cycle remains the same. Certainly also the improvement process can be improved.

2.2 Data Collection and Analysis

General data collection has started initially quite soon after business problem identification; target setting and scope definition have been done. At that time various options and material availability for initially chosen topics for example project management and release process practices have been reviewed.

Deeper data and material collection for different topics and from different sources started when Conceptual Framework has been created in draft level and when different areas for the conceptual framework has been chosen.

Big part of the information for the conceptual framework (IT project facts, software testing, project management and for alternative software delivery concepts) have been collected from the different sources through internet by participating for different webinars, collected the information from different groups and forums and whitepapers, from literature (Continues Delivery; Jez Humble and David Farley) and from people who have been interviewed in theme interviews or who have been reached by sending questionnaire.

Current state analysis for both current project framework and for current release process has been made by collecting the information from the current project. Once SWOT analysis was done it was reviewed with two people from the project. Some information has been slightly modified after review session to get the aligned topics and view for the project and release SWOT analyses.

Both qualitative and quantitative research methodology was utilized in this study. The qualitative research data consists of six in-depth interviews. Two of the theme interviews were done with project people and four of them with people who are working in international companies in IT-projects. Quantitative data research data was gathered with the questionnaire. Six questionnaires were sent for IT-project professionals in couple of international companies, four responses were received.

Theme Interviews have been made for six people, two of them are from the project and four of them are IT-professionals outside the project. People from the project are having the roles of product owner (project manager) and IT architect and people outside of the project all are IT professionals, three of them are either project/managers or one pre-present development view. All of them are working in international companies and projects are global. People outside of the project are coming from well-known, big companies operating worldwide. Theme interviews covered the areas about projects, project management and release and testing process related questions.

Examine Results and Result verification is done with the interviewed. These people are same who has participated result review session and who have been interviewed in theme interviews.

3 Current State Analysis - Project Framework

3.1 Current Project Framework - Process

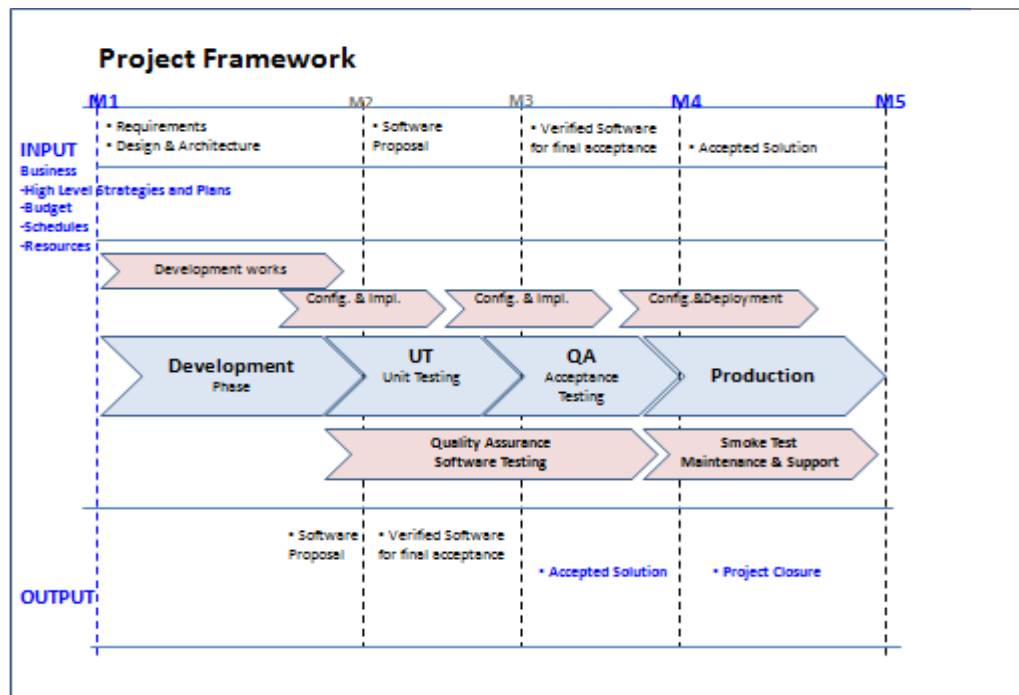


Figure 02. Current Project Framework - Process

In Figure 02 is shown current Project Framework - Process. Project management process is a process managing and controlling the project from project initiating/ planning phase to project closure.

When project plan and scope is agreed the release planning is done. Release notes - release scope is agreed FRS - Functional Requirements Specification.

One specific, agreed release is "a project". The project is closed after a sometime when it is released to the production. Timeline or duration when project is closed has not agreed. In the beginning of the project the full project for the first production delivery was in a scope.

Tools are same what are used in release process management. There are tools for Issue Management - JIRA and for documentation - Confluence. In Release process automation tool Selenium (IDE) is used and JMeter for functional and performance testing. Resources are allocated for the project from vendor side and in-house project

people are: product owner, business product owners, architect, developers and testing team. Product owner is the lead of the project.

3.2 SWOT analysis, Project Framework

Strengths	Weaknesses
<ul style="list-style-type: none"> * Product owner has clear role as a voice of the customer, he drives priorities, approve releases, makes use cases, he is a liaison between the development and the rest of the company. * "WaterScrumFall" process is developed * Continues customer feedback loop * Clear priorities for critical and major developments * Project Portfolio Management Tools are defined * Project Scheduling and prioritizing practises are in place * Current process and testing phases are defined * Resources and roles defined for the projects (releases) * Issue and Error management process defined * Change Management and Improvement (new features) practises are in place and works very flexible * Identified need for Measurement and Metrics * Training process practises planned and in pilot phase 	<ul style="list-style-type: none"> * Projects do not have baselines Time schedules are not followed and measured systematically Cycle Times are not followed systematically to improve the process * Full Traceability matrix is not in use from requirements to accepted solution No Systematic way to handle or follow up out-scoped features and what has delivered Measurement practises and metrics are insufficient in project management * Development and release plans from different vendors are not fully aligned * In "WaterScrumFall" process (Back-end) testing team starts the works too late * Project closure is not done officially * Too much documents and materials in confluence and it has not re-organized for process mode, original project materials are not separate place
Opportunities	Threats
<ul style="list-style-type: none"> * To re-organize project materials for continues process improvement structure * To develop security and performance testing practises * To standardise the Training Practises * To improve overall communication and information sharing about on-going and further actions, scheduling and what is agreed in project level 	<ul style="list-style-type: none"> * Lessons learned is not used after every release but continues customer feedback loop works in Front-End development * Improving the current practises is not done systematically or it is not visible for the project team * Cost Management and follow up is done only in high level and periodically reported

Figure 03. SWOT analysis of Current Project Management Process

Strength and Weaknesses are internal factors and Opportunities and Threats are external factors.

Strengths

Product owner has a clear role as a voice of the customer. He drives priorities, approve releases, makes use cases, and he is a liaison between the development and the rest of the company.

During the creation project has been developed "WaterScrumFall" process with the fast and continues loop for customer feedback and further implementation for back-end deployments is recognized.

Project Portfolio Management Tools are defined. JIRA is having a main role as a project portfolio management system. Different projects have been created for the system for different areas. There are projects for the back-end system (ERP), web-solution, partner channels etc. The tool gives the visibility to follow different type of issues in a structured way in the system. New features and Improvements are following the same practises.

In the continues project model and releases the scheduling and prioritizing practises are in place. Requirements from the different sources are prioritized based on the business needs by a key group of people. There are clear priorities for critical and major development works. Proposed time schedules for the half year or the year are agreed with the vendor. Agreed high level time and priority schedule is communicated for the different stakeholders.

Most of the needed processes and practises are created; project, release and testing practises. In different areas: project, release and testing management are identified development and efficiency potential. There are some examples to improve, for example processes: how those are represent, reporting: what additional type of reports are needed, what is the level of needed reporting for different departments and functions and what are the measurement practises and metrics in different areas.

Tool and process for Issue and Error management is selected and in daily use. Also Change Management and Improvement (New Features) practises and roles are in place.

Opportunities

Identified a lot of possibility to develop security testing practises.

Standardising the Training Practises is in pilot phase. Every time when release is going to be deployed in the production environment user training and training materials are given for the end users. Training session is improving the communication and information sharing related to the release and coming changes. In the end of the release training session could be short introduction about further and on-going actions and schedule of coming release. The short introduction session makes better the overall communication about coming things.

When everything what comes to the processes, reporting, measurement practises and metrics is been made from the scratch there is a huge amount of opportunities to continuously improve things. That gives very interesting works for employees who want to improve things and processes.

Weaknesses

Projects are not having baselines or clear measuring practises, for following schedule or deliverables.

Lead Times are not followed systematically to improve the process.

Development and release plans from different vendors are not fully aligned

Time constrains in all doings limit the speed up the systematic way of working in many areas for example in testing many things could be done more organized and systematic way.

IT Cost Management and cost follow up is only in high level and periodically reported to the top management. IT cost structure is not visible for people working on in IT department, not in high nor detail level. Clear templates and instructions how top management would like to see payback figures in investment calculations are not share with employees in IT department.

Measurement practises and metrics are in-sufficient in project management.

Threats

Lessons learned and Deming cycle not used to improve the process and project management framework.

Improving the current practises systematically not done or it is not visible for the project team.

Cost Management and follow up only in high level and periodically reported

Full Traceability matrix is not in use from requirements to accepted solution. Clear visibility from each of the requirement through the release process is missing. Tools and practises are implemented but repeatable practises are not visualized for the whole team.

There are not agreed practises for non-functional testing as a part of end-to-end testing. Performance and security testing practises and test cases in back-end release process are missing.

During the testing periods systematic testing wrap up practises are not agreed and Test Data management process is not implemented for all the Testing environments.

3.3 Current Development and Release Process - Back-end system

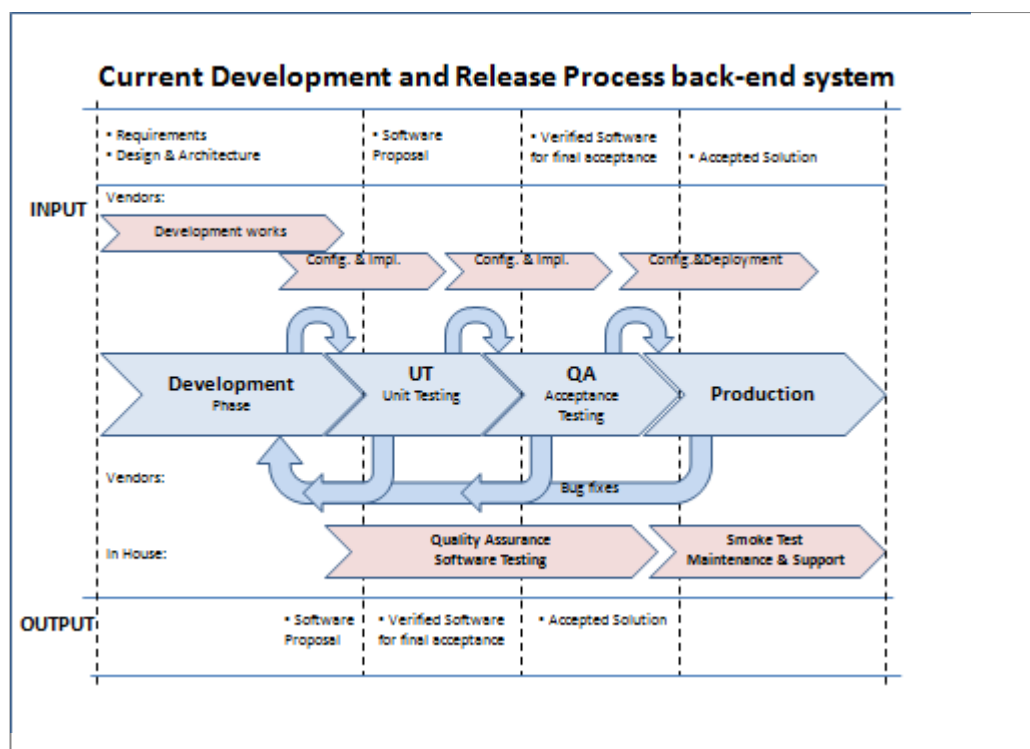


Figure 04. Current Development and Release Process

Release management is the process of managing software releases from development stage to software release in production.

There are three different phases before software release is ready to move to production environment for the real production use of end-user.

Before the development work can be started business requirements are gathered from the business side. Business stakeholders can be business decision makers who give the directions in high level, end-users who give the requirements for the usability and functionality or it can be partner channels or authorities from where the requirements are coming.

Business requirements can come also from the business strategies and from the higher level plans, for example the growth strategy has a direct impact for the backend system architecture, performance and capacity requirements. It is a business call how requirements are prioritized. When business requirements are prioritized and analyzed and specifications are done, design and architecture are clear and planned the development work itself can be started.

In current development phase many times also old bug fixes are included for the next release. There is not a separate process for the pure new development work (release) and for the bug/defect corrections. Once development work has reached the certain point that release or software proposal is ready for the further testing by the customer it will be implemented and configured for the testing environment, TEST environment.

In practice TEST environment is the first environment where unit tests, integrations & system testing are done. The full E2E scenarios (end to end) are run for the first time in TEST environment.

Test Data in TEST environment is not yet corresponding the one what is in production or in QA environment. In this testing environment services for external testing environments are available. Testing of the software proposals are tested by testing team. If any bugs/defects or process issues are found proposed software will be retuned back to the development team. Development work, in case of back-end software, is done by vendor. Vendor of back-end system is located in India. When the software/product proposal verification and validation is done successfully in TEST environment internally the verified software is ready to be moved for the next testing environment, QA envi-

ronment, for the final acceptance. Implementation for the QA environment requires currently manual configuration from the vendor.

Final acceptance of the software is done in QA environment mostly by testing team but also with help of the business end-users. The coverage of the testing: the testing scope and the testing cases; can vary when compared the same in TEST environment. The full functional testing what has been run in TEST environment is not necessary run in each release deployment because lack of back-end automated testing cases and the focus of testing in release deployment.

The testing focus in QA environment is more to simulate the new features and the production processes with the test data what is corresponding the one in production. If no bugs/defects or process issues are found in solution the final acceptance is given and the solution is moved to the production environment. Implementing the newly developed solution to the production environment requires currently manual configuration from the vendor. When the solution is configured in Production environment smoke test is done by testing team before the real production usage can continue. Deployment is done in Production environment.

When any type of the issue is identified in production environment it is analyzed what type of issue is and how urgent fix is needed. The systematic review process and practices for Production issues do not exist. Each found issue is analyzed and prioritized immediately. This practice gives the instantly the needed priority for blocker and critical issues.

Issue management for lower priority issues, both - during the testing phases or in production environment, does not have the systematic review and prioritizing process in place.

Tools are defined for release process management. There are tools for Issue Management - JIRA and for documentation - Confluence. In Release process automation tool Selenium (IDE) is used and JMeter for functional and performance testing.

The resources are allocated for the project from vendor side and dedicated testing team is in-house.

3.4 SWOT analysis - Current Release Process (Back-End)

Strengths	Weaknesses
<ul style="list-style-type: none"> * Project Portfolio Management Tools are defined * Scheduling and prioritizing practises are in place * Current process and testing phases are defined * Test levels and testing types are defined * Resources and roles are defined in high level * Issue and Error management and process defined: - tools and roles are in place * Test Management tool in pilot phase * Change Management and Improvement (new features) practises are in place * Measurement and Metrics needs are identified * Test Data management process is defined * Training process practises are planned 	<ul style="list-style-type: none"> * Release process Total and part Lead Times are not followed systematically to improve the process -Roles and Responsibilities not agreed in detailed level * Test automation tools are only partly implemented, for Back-End automation testing tools are not chosen * Test management tool is not in full use * Back-End Regr. Testing is a manual process * No systematic testing wrap up practises in place * E2E Process has dependencies on external systems in testing environments * Sprint/Release planning and management are used only in integration development * Full Traceability and matrix not in use from requirements to accepted solution * Test Data mgt process is not implemented for all the Testing environments yet. Measurement practises and metrics are insufficient if any
Opportunities	Threats
<ul style="list-style-type: none"> * Development potential in Regression testing * To develop different processes and practises for Release process and bug fixes * To share the automation test cases with vendor * To develop the test case library * Potential to further automate the processes in vendor sides (i.e. testing and configuration) -> Impact for total Lead Time * To develop security testing practises *To standardise the Training Practises * To improve overall communication and information sharing about on-going and further actions * Further development of Reporting * Automate partly the acceptance Testing 	<ul style="list-style-type: none"> * Speed of the actions to develop the process is slow * Limited resources to do process development works * Systematic way of working is only partly implemented in many areas * Cost Management and follow up is only in high level and periodically reported

Figure 05. SWOT analysis of Current Release Process (Back-End)

Strength and Weaknesses are internal factors and Opportunities and Threads are external factors.

Strengths

Project Portfolio Management Tools are defined. JIRA is having a main role as a project portfolio management system. Different projects have been created for the system for different areas. There are projects for the back-end system (ERP), web-solution, partner channels etc. The tool gives the visibility to follow different type of issues in a structured way in the system. New features and Improvements are following the same practises.

In the continues project model and releases the scheduling and prioritizing practises are in place. Requirements from the different sources are prioritized based on the business needs by a key group of people. Proposed time schedules for the half year or the year are agreed with the vendor. Agreed high level time and priority schedule is communicated for the different stakeholders.

In projects, releases and testing current process and testing phases are specified and the further development is on-going. Testing levels and testing types are defined in testing strategy. Resources and roles are described in high level for projects, releases and testing and deeper role descriptions are in the plans. Training process, timing and practises are planned and pilot is under work.

Tool and process for Issue and Error management is selected and in daily use. Roles are in place. Test Management tool is in pilot phase.

Test Data management process is defined and managing same way the test data from the production environment is current a practise in one of the testing environments. Same practise will be implemented for managing the data for the other testing environment in the short future.

Opportunities

Have been recognized that there is a development potential in Regression testing, still in many areas, for example in back end system, in Soap UI automation testing cases and in partner channel. Also partially automated acceptance test would give time and cost savings.

Opportunities are identified to further develop the release process. One option could be to separate different processes and practises for new development work in Releases and for bug fixes. Deeper co-operation with the vendor/-s would give the common advantages for all by sharing the automation test cases with vendor/-s.

When new release is deployed for the testing environment test cases are planned and created based on the release materials. In regression testing is used already created testing cases are used and needed adjustments are done for those. Organized and structured test script library is scoped. It would give time saving when everyone finds the right scripts for testing in scope.

While created the current practises during the last two and half years have noticed also the potential to further automate processes also in vendor sides (i.e. testing and configuration environment). Release total lead Time consists of all the steps what are related for the release process. If can reduce the partial lead time in one step it will reduce the total Lead Time.

Identified a lot of possibility to develop security testing practises.

Standardising the Training Practises is in pilot phase. Every time when release is going to be deployed in the production environment user training and training materials are given for the end users. Training session is improving the communication and information sharing related to the coming release and for the coming changes. In the end of the release training session could be short introduction about further and on-going actions and about the schedule of coming releases. The short introduction session makes better the overall communication about increases the awareness of the coming things.

When everything what comes to the processes, reporting, measurement practises and metrics is been made from the scratch there is a huge amount of opportunities to continuously improve things. That gives very interesting works for employees who want to improve things.

Weaknesses

In release process Total and part Lead Times are not followed systematically to improve the process. Systematically followed the dates and times what have spent for

testing in each environment gives the visibility where we are now and what is the lead time in every and each step.

Roles and Responsibilities are not agreed in detailed level especially in testing.

Test automation tools are only partly implemented and for Back-End automation testing tools are not chosen; only some example scripts are shared together with the vendor. Back-End Regression Testing is a manual process.

Test management tool is not in full use and piloting of it took too long time.

Resources are limited, daily actions and manual testing tasks take much of time. Speed of the actions to develop the process is slow because of very small amount of resources who are working to manage development works and production support.

Time constrains in all doings limit the speed up the systematic way of working in many areas for example in testing many things could be done more organized and systematic way.

IT Cost Management and cost follow up is only in high level and periodically reported to the top management and people working on in IT department it is not visible in high or detail level.

Threats

Full Traceability and matrix is not in use from requirements to accepted solution, clear visibility from each of the requirement through the release process is missing. Tools and practises are implemented but repeatable practises are not visualized for the whole team.

Sprint/Release planning and management are used only in integration development cycle not in back-end or it is not visible for the customer side team.

Full end-to-end process has dependencies on external systems also in testing environments. If external services are not stable it can cause delays for testing. Last step of end-to-end test with banks and accounting firm is not integrated for testing and there is not visibility what is tested and how in release process.

There are not agreed practises for non-functional testing as a part of end-to-end testing. Performance and security testing practises and test cases in back-end release process are missing in customer side.

During the testing periods systematic testing wrap up practises are not agreed and Test Data management process is not implemented for all the Testing environments.

4 Project Basics and Best Practices

IT - Business integration encompasses all the technologies that make the goals possible enterprise-wide for example information architecture, data management and integrations, application integrations and data management. The ultimate goal is a smoothly functioning IT foundation for business activities.

4.1 IT Project Facts

Be efficient and do cost savings are important actions in today's business. To full fill those actions it requires much from IT systems which are in more and more important role to run the successful business.

High expectations are expected and much is required from people who are running ICT projects, from project management. The great knowledge is required from both sides from business and from technical side. Overall IT architecture in one company can consist of many systems and synchronizing and managing the whole chain requires even more. Understanding the complex IT architecture it requires deep technical expertise.

This gives also requirements for delivering products in more and more efficient way and therefore efficient, low cost concepts, tools and methods are more required; especially in the bigger IT projects or in complex architectural environments.

Following examples give a great perspective for the current situation in ICT world and tell the possible reasons why IT projects might fail or what could be the things to avoid failures. What could be the actions to take and how to succeed in IT projects?

Computer weekly published some news about how 'Big businesses are collapsing because of failed IT projects'. In the article Flinders refers to the research done in Oxford University that IT projects are 20 times more likely to run out of control than other busi-

ness projects. One in six IT projects run out of control with average cost overruns of 200%, (Flinders, 2011).

The concern in the article was that for example some government projects are focusing only on average performance. This can cause large overruns and delays.

Nowadays project business sponsors are focusing on what is the real business impact. Understanding the direct business impact of the project on the other department and the overall organization are also the required skills from project leaders today. Projects are more complex with several applications and efficiency is mandatory in business processes and in work environment.

Projects should be strategically aligned with the strong expectations giving and building new capabilities and maintaining the competitive advantage in a market place.

4.1.1 Why ICT projects fail?

There are different public data sources and articles indicating about similar type of reasons or areas what can cause failures or in which area to consider doing things better or what can cause delays in IT projects. TOP ten lists are including the following reasons in different orders or listing only some of the topics.

Following list is adapted from the Centreline Solutions Inc. Presentation already from the year 2004 and 2005 but including still same facts what we are also facing today in the projects.

10 Major Root Causes for the project failures are:

1. Change Management, 2. Poor Communication, 3. Insufficient / inadequate Resources, 4. Poorly defined requirements, 5. Inaccurate estimates, 6. Poor Risk Management, 7. Poorly Defined Deliverables, 8. Over Optimism, 9. No time for Project Management (including roadmap and controlling) and 10. Improved Project Management skill set is needed, (Centreline Solutions Inc., 2005; V02.18).

Quite often projects are squeezed at the end, projects can be stressful and in many cases have to deal with unrealistic expectations or customers could feel that projects are out of control. In the document is very valuable small picture what encapsulates the scope and related aspects.



4.1.2 Project measures and metrics troubles J. Kendrick (PMP, CMC)

Example of project measures and metrics troubles is adapted from the presentation J. Kendrick (PMP, CMC) Project Management Office Summit: Measures and Metrics for PMO success, 2009 P2C2 Group Inc.

Most organizations have problems with their measures and metrics

63% of projects have schedule delays

49% of projects exceed budget or do not meet business objectives

45% of projects face cost overruns

23% of all projects FAIL

WHY?

- Too much data
- Measures aren't useful
- Too much time to collect and report data
- Data quality is poor
- No baseline
- No time to analyze data
- Reports aren't used for decisions or action
- Metrics don't document the PMO's accomplishments
- Executives don't pay attention

4.1.3 CIO - Pearl Zhu - Ten Root Causes of IT Projects Fail

Pearl says in the article in Future of CIO (Pearl, 2014) that every IT project is the business project and must be planned well, "if you fail to plan you must plan to fail", need remember the 5 P's:

Proper Planning Prevents Poor Performance (90% planning, 10% implementation). It is good to keep in mind in the projects that "Probably 90% of application project failure is due to politics!".

Shorten the Ten Root Causes of IT Projects Fail according to the CIO forum are:

Business purpose of project is not well-defined and communicated, **Business Plan** is lack of structure or detail (project management model, technical solution, resources, skills), Poor **Planning** (not allocating time to design, testing and requirements gathering), Lack of **Executive support**: compete priorities across various stakeholders (conflicting priorities), Poor Execution (humans in the process, Lack of **Quality Project Management** not follow the 3C Rules (communication, coordination & collaboration), Over Specification - it is good to remember that operational processes will change once a new system is introduced, Failure to **Adopt and Adhere** to a suitable **Project Methodology** – this is very important to remember because if using waterfall where business has an agile mentality or if using Agile without a full sense of the overall plan/roadmap could lead to failure, Poor **Change Management**, Technical Challenges (IT department is adding cost and complexity and the project exceeds the cost / benefit), Governance was missing or was skipped (technical delivery of the software, not delivery of a business solution - business participation), missing the project management process components (standardization project management processes) and integration with other corporate processes (e.g., procurement, strategic planning). Prioritization of projects, utilization of project portfolio techniques and continuous improvement are the key things to follow for successful implementation.

Additional for that very interesting could also be: Both business and IT are to blame for project failure. Successful IT projects organizations do need establish a project management framework with four core elements like processes, organizational structure, people and systems and a effective project management methodology is required for managing scope, time, resources, change, risk, cost, issues, configuration, quality and communication .

4.1.4 Why IT projects fails - IT Business Edge?

There is an interesting power point presentation of Why IT projects fail based on IT business edge telling the root causes and reasons what are makes uncertainty and confusion in IT projects making them to fail.

Some main root causes are **Lack of governance** in Project criteria, roles and processes. Outcomes are not established and actively monitored. Lack of management support is the reason for many failures in different areas, management support should affect in all levels in project organisation. The other reason what makes communications and open atmosphere challenging is **Internal Politics** in the projects. In many projects there are politics on the background and it mirror in daily actions. Essential things are open communication and focus on the right things, focus on the project's overall value internally in the project team. The other aspect for communications comes from what is happening between IT and Business side. Many times in the projects is heart about **poor communications between business and IT**. One solution for poor communication between IT and business is that IT people really need to try to understand what the business requirements are and what business people are telling them. IT people are in key roles in the project.

Unclear Business Requirement makes unclear expectations

In order to avoid ambiguous expectations in a project it is necessary to understand and fully outline requirements and communicate clearly for stakeholders at the start of the project. Projects produce the solutions what are based on business requirements.

Lack of Fact base analysis: Facts are facts and assumptions and opinions are entirely different things. According to Nigel Hughes who is a global Market and Service Development Director is saying that project ' plans are not based on the facts but on assumptions and opinions' that type of planning is not keeping up good spirit and project progress. Plans based on assumptions and opinions help the projects to lose potential benefits what well manage project can give for the business and stakeholders. Mr. N. Hughes suggests project analysis to include basic metrics such as 'cost efficiency, productivity, cycle time, error rates and so forth.

Lack of input from users can cause that in some projects is possible developed the tools or solution what is incomplete or even unnecessary for the users. Input should come from the users.

Changes in Project size, budget and scope can be daunting task for many project managers if they don't have the right tracking tools implemented. Tools are needed to keep on track project follow up, change requests, costs and risks. Further ´more about some of the tracking tools available in the market.

Changes in key personnel: The project members make the project progress. There are also tools in the market to help cross-functional collaboration, how key stakeholders, project manager, project teams and subcontractors collaborate. One example is cross-functional flowchart provided by Michael Tylor.

Overrunning schedules can be caused by unrealistic promises to project owners and stakeholders. If project owners and stakeholders are already committed for the unrealistic schedule one try to catch up the schedule is adding resources but many times it does not solve the problem.

Higher than expected maintenance fees: Even though everything is done in the projects to satisfy the business needs there should be a nominated business owner for the project who can substantiate the value for the project. Mr. Mike Sisco, the president of MDE Enterprises, IT Management and company, says: "If I can't find that, I'm going to cancel it, even its 60 percent done".

Inadequate skills of project completion: Project Managers should know exactly what skills and expertise are available within the project team. When they know in individual level the skill sets they are able to recognize the knowledge gaps if there are any. By this they can help the project to avoid possible failures.

4.1.5 IT Project Success Factors

There are few different factors what are critical success factors for the IT projects. The following example is adapted from the GTNews publication (Afponline.org, 2014). The slide is covering the project success factors from the four different angles:

Customers, Processes, People and Technology.

In the slide the whole core and the interaction of different areas are clearly expressed.

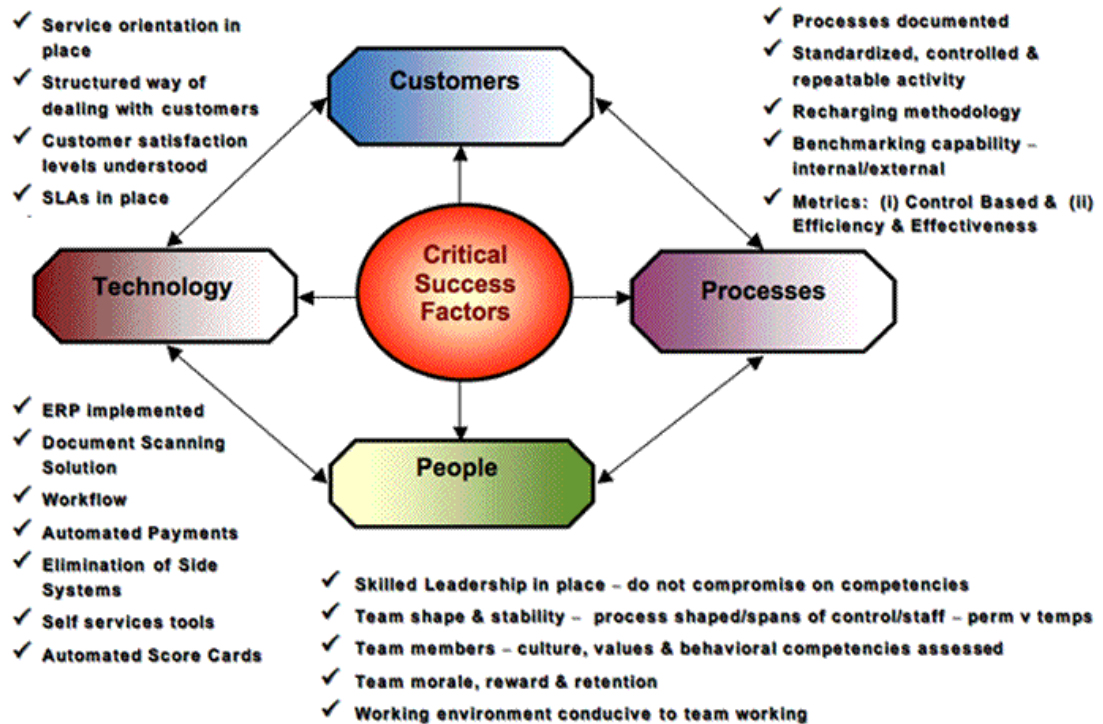


Figure 06. Critical Success Factors adapted from GTNews publication (Afponline.org, 2014)

There are few interested sources and consultants presenting the success factors divided for the main functions in core areas. Success factors can be presented in project and product level or taken a helicopter view and presenting the success factors in a higher level depending on the viewpoint of the presenter.

Colleen Winkler presents (Winkler, 2010) the list of project success factors and percentages of response (viewpoint is purely project level):

Project Success Factor	% of Responses
1. User Involvement	15,9 %
2. Executive Management Support	13,9 %
3. Clear Statement of Requirements	13,0 %
4. Proper Planning	9,6 %
5. Realistic Expectations	8,2 %
6. Smaller Project Milestones	7,7 %
7. Competent Personnel	7,2 %
8. Ownership	5,3 %
9. Clear Vision and Objectives	2,9 %
10. Hardworking, focused personnel	2,4 %
11. Other	13,9 %

Martin Bauer presents the success factors in the article How to Plan CMS project (Bauer, 2008). He is pointing out that the client / customer matters and more clients a slightly more complicated it becomes to meet all the customer expectations at the same time.

According to Martin Bauer the success factors are: having satisfied stakeholders, meeting the project objectives/requirements, meeting an agreed budget, delivering on time, adding value, meeting quality requirements and sense of professional satisfaction for the team.

4.2 Project Management

4.2.1 Project Management: Foundation and Planning by Karl Wiegiers

Wiegiers (2014) introduces valuable practises which are organized into five categories.

1. Laying the foundation
2. Planning the project
3. Estimating the work
4. Tracking your progress
5. Learning for the future

His advice is that every time when initiating a new project builds the corresponding activities for the project. Those activities which add value for the project, only those activities to be applied.

1. Laying the foundation Define Project Success Criteria

At first, right at the beginning of the project stakeholders should have a common understanding about project success, business interest and expectations. The project should also have clear and measurable goals. Based on the business go. Commitments should be negotiated.als the project could have achieving schedule and budget targets and clear priorities. In every project there should be defined the clear release criteria's. Whatever the criteria's are selected they should be realistic, objectively measurable, documented and aligned with what "quality" means to the customers.

2. Planning the project and write a plan.

The time what is spent analyzing what it will take a solve the problem will reduce the number of surprises in the project. Useful plan includes lots of things to be considered: resources, budget and estimated plans for those. Teams are distinctly defined and responsibilities are determined.

Teams and other resources are acquired and training plan is available. In the project plan and schedule the major deliverable dates are determined. Software delivery cycle is identified and project will follow it. The plan is done how to track, monitor and measure the project and which metrics to use. It is decided how to manage the sub-contractor relationships. Planning the project it more than a task list for the resources.

Good plan includes the planning sheet for the common large tasks. The plan contains the plan for the rework after the quality control activity and risk matrix or corresponding tool to control project risks. It is important to identify the risks, evaluate the treats and make a mitigation plan. Planning part should have planned time for process improvements. Allocated time for the improvements will start to reduce the time what will be spend for the same activity next time.

3. Estimating the work - Estimate based on the effort

Many times people provide estimates in units of calendar time. Karl Wieggers prefer to estimate the effort in labour-hours associated to the specific task and then translating the effort into the calendar-time estimate. Quite often only about fifty to sixty percent of the work hours can be spent as an efficient project hours. In order to be more efficient clear priorities to be set for the project team members and let them then focus on one or two objectives at a time.

Scheduled time for the trainings is important. In high-tech, fast changing area people need also time for educating themselves. In some cases training can even be a team building experience and people can learn how to apply the same improved practises to their common challenges.

One way to improve the estimation process is to record the estimates and how to derive estimations. Understanding the assumptions and approaches what been used to create the estimate will make defend and adjust them when it is necessary needed to be done. This practise will also help to improve the estimation process.

In the article is introduced shortly The Wideband Delphi method which is an efficient group estimation technique. The method is more reliable than just asking people the best guess. In Wideband Delphi technique small group of team experts are asked anonymously generate individual estimates through iteration.

4. Tracking your progress - Record actual and estimates

Each individuals can begin recording the estimates and actual and project manager/lead should track this project data items on a project task or milestone basis. Project manager could estimate and track the size of the product in terms of requirements, user stories, line of code, function of points, GUI screens or other units make sense for your project some examples could be for example development tasks (web), new development / new feature or functionality). Planning the tasks in more detailed level will help on that.

Project status to be tracked openly and honestly. If projects are planned one day ahead there is no visibility for any coming activities and project manager do not know how far behind of plan the project really is. More and more surprises start to appear.

When the Project Manager has build the atmosphere that is good and safe to report accurately the project is also run and traced more accurately. Better practise is to run the project bases on the data-based facts rather than from the misleading optimism.

The project can be managed more efficiently when it is really known what is done and what is not. Then the project manager only know what things are falling behind their estimates and why and what problems and risks there still are and to be tackled.

It is good to define few areas of software measurement, for example five metrics (size, effort, time, quality and status).

5. Learning for the future - conduct project retrospectives

Retrospectives (post-project reviews) will help the project team to develop the process improving the done activities. In retrospectives after every project, project phase, release or iteration lessons learned are collected. During the review session team identifies what have gone well and what can be improved for the next time.

4.2.2 Transforming to Agile Project Management

The presentation and result from Kendrick (2009) shows that there are few important things in project management to consider. They are: Measurement Levels Strategic Goals, Strategy Map, Balanced Scorecard and/or Strategic Indicators, Program Management, Portfolios and Programs, Projects and Operation and Maintenance.

Map Measures & Metrics in Projects are expressed Monitoring Project Time (Time Management Performance, Actual vs. Planned Duration, Total Time to Complete (by phase by task), Comparing Costs (Actual vs. Planned Costs, Original Estimates, Base-line changes), Measuring Product / Service Quality (Availability, Customer Satisfaction, Error Rates), Measuring Risk Management (Risk Indicators, Incidents, Impact on Budget, Schedule or Quality) and Corrective Action Plans (Project Initiative to Fix, Effectiveness of Corrective Actions). Very valuable blogs, comments and examples are available in IT Knowledge Exchange forum.

One example for Quality Assurance and Project Management is measuring Project Budget, Project Composition, Project Estimation, Project Management, Project Resource, Project Team, Software Development, Software Project, Software Release and Team Composition from cost and time wise.

4.2.3 Project Manager's Agile Survival guide by Rally Software

Rally Software (2014) has published an interesting article about Project Manager's Agile Survival guide. The main message in the whitepaper is that agile project and so called traditional project are not that far from each other's and all the roles and practices can be mapped for corresponding roles and tasks.

Agile refers to practices that encourage rapid deployment of working products, focus on quality and value and integrated cycles of feedback.

The results of a global survey about project management practices shows that according to the survey 34 % of project managers are using Agile PM methods in the companies and interesting information is that 62 % of Project Managers in those companies are Agile practitioners. The full mapping is expressed in Appendix 3: PMBOK Practices mapped to Agile Practices by Rallydev.com.

4.2.4 Mapping of Waterfall Project Management to Agile Practices

Rally Software (2014) depicts that mapping of Waterfall Project Management to Agile Practices is not the huge task and Agile Practices can help on managing the complex work and improve project delivery, agile projects are iterative and incremental, feedback loop is fast. Agile and Waterfall development aren't as different as people image they are. There are differences in these two approaches in the implementing costs, schedule and scope what both are recognizing.

In the waterfall projects scope is freeze requirements and based on that schedule and costs can be planned and in agile project scope is recognized as variable and teams are given the tools what are needed for the success. Teams get time and space to complete the work and command-and-control management disappears.

In the traditional projects Project manager is responsible person about scope, costs, schedule, quality and reporting and in agile way the whole team shares the responsible having the same goal /target. The whole team is engaged and motivated contributors who produce high quality work at a faster pace.

Agile Methods are just as compliant with the Capability Maturity Model Integration (CMMI) as traditional waterfall model. Agile project phases map nicely to iterative development environment. This means in practice scaling from short iterations up to longer-term releases (Rally Software, 2013, p.9).

The other example is how Planning process in Traditional Project Management is mapped the same in Agile. In Agile Planning Process Agile team produces high quality work at a faster pace.

In Appendix 4: Mapping Integration Management to Iterative Process in Traditional and in Agile Practices can figure out that corresponding phases are in both processes. Planning Phase in Traditional mode consists of Developing the project chart, schedule, a Plan and in Agile the corresponding planning phase get Roadmap and Backlog created. Then follows the execution and in Agile side it is called Iteration work. When in traditional projects the project is monitored and controlled by Project Manager in Agile way there is a person who facilitate, lead and collaborate the project works. In traditional project there is an integrated Change control and agile side constant feedback and ranking backlog can be recognized.

In different organizations agile way of working can be adopted at a speed that works best for them and agile practices can be customized. Utilizing the agile in portfolio level changes is needed in organizational level, for example in processes, structure, roles, reports and metrics.

In all cases the shift to Agile project management it is a cultural shift. Project and Program managers are having an essential role assisting in identifying the right platforms, processes, project statuses and metrics for delivering the useful information to executive management.

Agile project facilitator has an important role of facilitating self-managing teams and guide them with the right questions in a right time to proceed towards high-quality, collaborative work and being proud of their achievements.

The first step in transforming to Agile project management is to learn the Agile process. Rally Software represents some agile materials to study the agile methods and agile project management. For the basics can read The Agile Manifesto from agilemanifesto.org. In the manifesto can be found origins, principles and how founders came to develop new approach to development work.

4.2.5 "Subway map" by Agile Alliance

"Subway map" illustrates very logical way the different areas of concerns (Extreme Programming, Teams, Lean, Scrum, Product Management, Design, Testing, Fundamentals), Agile Alliance (2013).

4.2.6 Kanban: by David J. Anderson

Kanban system, continues improvement culture, implementing and coordinating Kanban, Issue management and metrics are key drives in successful Project Management.

Focus on Quality

Improving the software quality is a well understood problem. According to this article both agile development and traditional approaches to the quality of work and these two approaches should be used in combination. Excessive defects or software bugs are the biggest waste in software development.

To get the better quality and the better coverage to the testing one option is Test Driven Development (TDD). Test Driven Development does seem to provide the advantage that coverage of testing is more complete, (Anderson, 2010, p.28).

Reduce Work-in-progress and Release often

The following figure 12 is directly adapted from the document. The graph in the figure is showing the Time what is spending from the requirements analysis to the release for the testing. Based on the shown graph has been suggested that there is a correlation between increased cycle time and poor quality. It seems that longer cycle times are associated with significantly quality, (Anderson, 2010, p.29).

Cumulative flow diagram depicts the quantity of work in a given state. Different phases on the flow are explained as follows:

- inventory is backlog
- started implies the requirements explained for the development team
- designed means developed for the feature
- coded means have been implemented
- complete means unit tests are done and released for the testing team

The first line on the graph shows the number of features in scope for the project. In this example the scope arrived in two batches from the business owners, (Anderson, 2010, p.28-31).

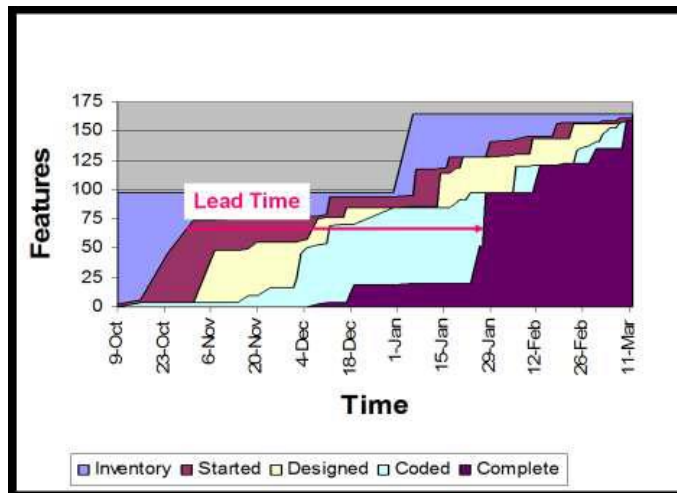


Figure 08. Cumulative flow diagram Time by Amount of features (Anderson, 2010, p.31)

The vertical number between the second and fifth line is indicating the quantity of work-in-progress, while the horizontal distance between the second and fifth line is showing the average cycle time from starting a feature on that day until it was finished.

The horizontal distance is an average cycle time.

Reducing work-in-progress or shortening the length of iteration will have a significant impact on initial quality in development work. Defects will rise disproportionately to the increased quantity of WIP.

There are some experiences that two week iterations are better than four week iterations and one week iterations are better than two weeks. Shorter iterations / lead time from requirements to release in testing could improve the quality of development work. The following logic indicates that could make sense to limit WIP by using kanban system.

Small releases

Small releases show commitment from the development team and build the trust with stakeholders like operations, technical support, and field engineering and sales. Small releases show that the software development team can deliver and is committed to providing value.

Prioritizing

There has been lots of discussions who should do prioritizing and how it should be done. Option introduced in the document is to have the product owner, business spon-

sor or marketing department to improve prioritization of requirements. Engineering management has a very important role to influence how the prioritization is done.

Measuring the success

The ultimate measure of success is business value delivered. Organizations should be able to measure and report the business value delivered. In order to achieve this the capability maturity ladder to be builds and the team must climb.

Kanban enables incremental changes

Kanban enables incremental changes and reveal opportunities for those improvement which do not involve complex changes to engineering methods. Significant improvements are only possible by managing bottlenecks, eliminating waste and reducing variability that affects customer expectations and satisfaction.

Continues Improvement Culture

There are successful companies in which continues improvement culture is part of the workplace culture and the entire workforce is focused on continually improving the quality, productivity and customer satisfaction.

For the right time decision making the right people are needed for the right places; the roles and responsible are defined.

Things to be considered are which items are ready to release to the production, what is required to release, what are the risks and what is the mitigation plan, who are needed and when exactly, what is needed when released in production. Outcome should be the complete template representing a release plan.

Making On-Demand and Ad Hoc Deliveries

Regular deliveries have advantages. Making a promise to deliver on specific date allows those who are involved schedule it. It provides certainty. This activity reduces coordination costs because no overhead involved - all that is established once and is fairly consistent form them on.

On-demand and / or ad hoc delivery makes sense when release activity costs are small. In the document is said that "When coordination costs are low then there is no benefit in scheduling coordination activities regularly" and " Secondly, it makes sense

when the transaction costs are low, perhaps because the deployment of the code is largely automated and the quality is assured in advance prior to deployment", (Anderson, 2010).

From process wise it makes to make sense regular activities, it makes process faster and everyone know all the time, on time and on line what is happening. When process is enough efficient and fast then "And finally it makes sense in environments when the deployments are so frequent that there is no real need for a pattern to be developed. New software is being delivered so often as to appear continuous to most observers and external stakeholders."

Who is Mr. Deming?

W. Edward Deming is one of the three main fathers of the Quality Assurance movement of the 20th Century. Mr. Deming has evolved the use of Statistical Process Control (SPC) and he has deployed it into a management technique he called the System of Profound Knowledge. This system intended to prevent managers to make poor quality and/or intuitive decision and replacing poor decisions with statistic based objective and counter-intuitive decisions, (Anderson, 2010, p.157).

Deming cycle also called PDCA and plan-do-study-act (PDSA). Cycle is used as a model of continues improvement, when developing a new improvement project, developing a new or improved design of process, product or service. Plan-Do-Check-Act is a procedure used for plan (recognizing an opportunity and plan for the change), Do (testing it), Check (review) and Act (actions taken based on the learning, (ASQ, 2014, Quality Tools, Learn about Quality).

Invest for Automation

Anderson says (2010) that Natural strategy for capacity elevation is investment to automation. A wonderful side-effect of automation is that it is also reduces variability.

Transparency

Transparency in the kanban system will help to recognize capacity constrains (bottle-necks) in resourcing and if there are any issues impeding flow at that point in the system.

Critical performance improver is usage of kanban transparently tracking and reporting projects. "Transparency leads to visibility of both bottlenecks and impediments and

consequently to improved exploitation of available capacity to do valuable work, through a team focus on maintaining flow.", (Anderson, 2010).

4.2.7 Five challenges to Agile Planning

Jama Software has released an interesting article about five challenges to Agile Planning. Author of the article is Dorian Simpson, Managing Partner, and Planning Innovations. Dorian is the person who has led the successful developments and launches of innovation products and services more than 20 years. He has been leading the development of the world's first 500-channel digital satellite system.

The whitepaper is advisable article addressing five of the major challenges what Jama Software lead to Agile failure, also giving advices how to get the Agile hum in the organizations.

Even most of the Agile projects developers would like to start the works immediately there must be given sufficient time to provide the business requirements what is the basic for the product and for the project. Business requirements in higher level give the directions for the projects and also for the development teams what is expected from them.

What is the sufficient time for business requirements? It does not necessary need to mean months what is sometimes be seen when collecting the requirements.

Often it takes anyways weeks to get it started. There is a need to create a vision, product plan and prioritizing the use cases, preliminary design, market analysis and so forth. Development team is involved for architecture, platform and performance planning.

Product Owner has a leading role in orchestra, the decision-maker. He / she is "the voice of the customer". He/she drive the priorities, approve continues software releases, write use cases, approve them and be the main contact person between IT and Business organization. He/she cannot drive his /her own ambitions or personal opinions into the product. Product Owner role is hands-on role and requires fulltime and full commitment.

The important role also is Product Manager's role. Product Manager is the person who participates in planning meetings, agree the priorities and implementations but leaves the approving practices for Product Owner who is more technical person.

In order to involve the real customer / end-user of the system the target customer group to be identified because collecting input from the real end-users is slow, in some cases input is irrelevant or they just do not know how and what.

Early adopter customers would like to be part of the developing new things and part of the success. Once the direct and rapid access to customers is build the skills actively listening them, launch surveys and ask them what they want, like and would like to have. Learn how the customer is thinking. Communication is in the important role again. Use cases to be created and prioritized. Explaining the new feature, value of it and creating the test cases will open up the requirements for the developers, (Simpson, Jama Software (2011)).

4.3 Software Testing

4.3.1 Pre-conditions for software testing - Entry Criteria

Before it is reasonable to start testing there are some pre-conditions to consider.

Without taking any deeper sight to the every topic in the entry criteria, the main topics are listed: Test environment (set up for environment, configurations, interfaces), roles and accounts are defined (correct roles with access rights), test planning is done (test strategy, test plan, test scenarios/cases), previous tests are successfully done, no open critical issues and test data is defined/available,

4.3.2 Software testing in general

Software Testing is a set of activities what are performed to verify and validate the software against the requirements. Is the software built right (according to the specifications) and is the right software built (according to the customer requirements).

According to StarkInfoTech (2014) Software Testing Life Cycle (STLC) model is identifying what Testing activities need to be executed and what ii the best time to perform the activities.

There are different models introduced for describing and modelling software testing life cycle (V-model, testing process flowcharts covering a bit different or in the other words said testing steps) but typically software testing life cycle consists of few testing related steps from test planning to test execution. One source, StarkInfoTech introduces software test process with the following steps: Test Planning, Test Analysis, Test Design, Construction and Verification, Testing Cycle, Final Testing and Post Implementation.

In practise there are different testing levels in testing cycle and those are different type of testing in different testing environments, for example Unit Type of Testing in one testing environment, Functional and non-functional and end to end business process testing in production corresponding environment and smoke test type of testing in production environment itself. Different type of testing can also be done in a different way depending on the project and approach. In some cases also approach can vary by project and scope.

Before Software Testing Process can be started Requirement Analysis should be done.

Software testing can be seen as a part of Software Testing Life Cycle. There are different testing approaches like manual and automated testing and non-functional requirement testing. All these approaches can be included in a testing process.

Different steps or phases in testing process are described in details. The first thing before Software Testing Process can even is finishing requirement analysis. One approach to finishing the requirement analysis is that testers analyze the customer requirements and work with the developers to see how and what requirements are testable. This will happen in the design phase. Before that can happen or parallel on this requirements should be clear and agreed with business side.

It is important to start test planning and unit tests as early as possible to find the gaps or issues. Later the issues found more expensive it would be.

In the test planning phase test strategy and more detailed level test planning is done. Resources, risks, costs and time schedules are planned and communicated to the stakeholders.

In test analysis and test design phase detailed level planning and analysis is done for the testing. In this phase can be planned what part of testing and test cases can be automated, what are executed manually, what non-functional testing will be executed

and how much time is needed for each of the area, what can be done parallel and what resources are used for each of the area and each of the testing phase. In test designing can be included black-box and white-box test techniques. Manual and automation test script creation can be included in the test design phase.

Test construction and verification phase to be completed detailed level test plans and testing cases also non-functional test plans to be ready.

Typically test cycle includes test execution and bug reporting. Test results should meet the plans. There can be one or several test cycles.

Depending on the schedule, priorities and bug severities test cycles can be closed or prioritized and the product under test can be moved to the production for the use of real business users. Before moving to the production final tests and the decision to be done to move to the production. Needed documentation/evidences about testing and test results to be available for the decision making. Roll-back plan and actions to be taken before moving the product to the production.

In post implementation phase production smoke test will be done. Lessons learned will be collected to avoid the faced issues in the future releases.

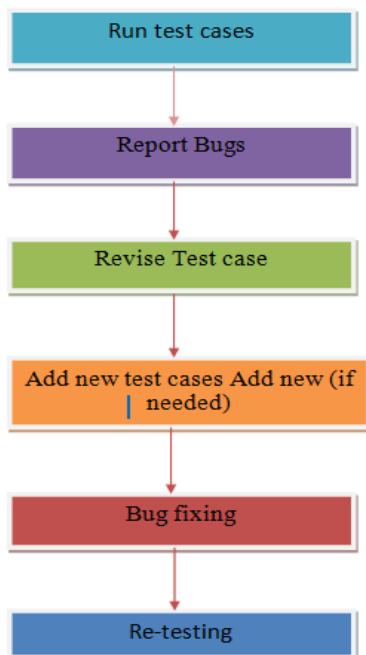


Figure 09. Testing cycle by StarkInfoTech

Verification and validation

SoftwareTestingSoftware (2014) defines that verification is answering the question Have we built the software right and does it match to the specification? Validation is answering to the question Have we built the right software and is it what customer wants?

Software testing is validating and verifying the software / product.

Software Testing can be started when development work is in a mature level that it is reasonable to start testing. Test Strategy and Test Plan are the documents explaining chosen testing strategies and methods, approach and entry criteria and selected tools in a detailed level. Test Strategy and Test Plan documents also have information about scope, chosen techniques, tools or methods, responsibilities and Issue Management.

Software testing ensures that the software under work will meet the requirements that has guided through design and development, work as expected and can be implemented as it is testing is completed.

Different software development methods have different testing efforts and testing points in the development process. For example in agile method test driven development has pushed much testing for the developer before it reaches the testing team. Software testing starts at the same time or moment the project and development work starts and continue till the end of the project.

Quite common practise also is that testing is performed by a group of testers only after the development work is finished. This approach often compromises testing as a buffer time in the project and if the project is delayed testing time is reduced.

Software testing provides stakeholders the information of the product quality and maturity to move new features or the product to production environment or for the public sales.

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not

limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects), (SoftwareTestingSoftware, 2014).

The Software testing topic is a huge area to cover but in order to raise the importance of functional and non-functional testing (performance and security testing four topics are selected for a bit deeper inspection: regression testing and utilizing the automation help on that and performance and security testing.

4.3.3 Manual Testing



Figure 10. Manual Testing cycle (SoftwareTestingSoftware, 2014).

Manual Testing is the process of executing software to validate against the requirements. In manual testing aim is to find the software bugs / defects before release in production. Manual testing is done by human being not by the machine / automated tool or by script. There are different phases in manual testing for example unit, integration, system and user acceptance testing, operational readiness test, alpha and beta testing and smoke testing. Manual testing may involve various interfaces like GUI (Graphical User Interface), CLI (command line interface) and /or automated frameworks to test APIs.

Manual testing typically involves using Black-box, White-box and /or Grey-box testing methods depending on what type of software testing is performed.

Manual testing can be performed as Statistic or Dynamic testing. Statistic testing means testing software against the requirements, verification and dynamic testing refers to execution of software uncover defects / software bugs.

Manual testing is performed by testers, in different testing phases it means that for example in unit testing developers perform unit tests, testing team can perform manual testing in many different testing phases and for example end users perform manual testing in user acceptance testing phase.

Testers can use test plans, test cases or test scenarios to help the test execution or they can do more exploratory testing without separate written test cases.

4.3.4 Regression Testing and Automation

Regression testing as selective software retesting for those software parts that have been modified (bug fixes, newly added features) since previous versions of the software. Regression Testing is initiated after programmer has attempted to fix a recognized problem or has added source code to a program. Newly added code should still comply with its specific requirements and unmodified code has not been affected. These can be measured by quality control measures, (Wepopedia, 2014).

At any time software is implementation modified regression testing is needed.

In Regression testing is tested that bug fixes are not impacted for other software functionality of the application and the software is working as it is expected to work before the code changes are done. Regression is generic and it is not necessary specific for any bug fix or code change or involve those changes but only execute the separate regression testing cases.

Regression testing can be done by rerunning existing tests against the modified code whether the done changes have broken anything that worked prior. The focus in testing is on finding defects after the code change is done. In regression testing main processes are retested and ensured that existing functionalities and working and old issues have not come back again.

Efficient regression testing requires well-developed test suite and automation in testing. Automated functional GUI testing tools are used to repeat system-level tests through the GUI.

When changes against the existing functionalities are regression tested, new testing cases typically become as "regression testing cases and to be added in regression testing case library or existing regression cases to be modified including the newly introduced changes.

There are some strategies and factors to consider during regression testing. Here are some examples:

- Test bugs and functionalities around them carefully.
- Check that related processes and components are still working.
- Focus on functional issues at first

Building a Regression Test case Library

The most effective approach is to build up a regression test case library. Every time when there is a new build or change in the software regression tests (test battery) will be run from the library.

Keep the library tidy, avoid duplicates and review and check the test cases periodically. It is possible that when there are many people updating test cases also duplicates appear time to time (Microsoft Developer Network, 2014).

There are examples when teams avoid performing regression testing and focusing only for the modified part of the software.

Every time when software is modified and functional tests, unit tests, integration tests, and build verification tests are successfully done regression tests can be incorporated to the testing library.

Automated testing is typically required

Guidance is to spend as less as possible time with manual regression testing, Regression testing should be automated.

There are various automated testing programs that make regression testing relatively easy to manage, couple of clicks of a mouse to establish set of testing parameters and to check new iterations of code against previous software version. Inconsistencies are checked from the logs and specified where the code broke and why.

In most cases there isn't time to run the entire software through but coverage is dramatically better than doing testing manually.

Regression testing to be seen effective and cost-efficient, comprehensive part of the testing strategy and test planning. Regression testing is an essential aspect of a dynamic, iterative development and deployment schedule. Well-planned testing strategy with automated regression testing at its core, project keeps the planned budget, team is kept on track, unexpected bugs are prevented to damage the project.

4.3.5 Performance Testing

Performance Testing can serve different purposes like demonstrating that system needs performance criteria, comparison of two softwares and how they perform or it can measure which part of the software can cause the performance issues.

Performance Testing is validating non-functional requirements (response time and throughput) and validate software quality (reliability, scalability and efficiency). Performance testing has become vital for example for web-applications. Those components that affect for the performance goal (response time, throughput and scalability) are called performance bottlenecks. Objective of performance testing is to find the bottlenecks and to retest when performance fixes are done.

There can be many type of issues causing performance bottlenecks, some examples: frequent updates to log files or database tables, not enough hardware resources for running software, network performance issues, database is not indexed, database and SQL queries have performance issues.

Performance Testing Types are for example Load Testing (checks the behaviour of the software under normal and peak load conditions) - conducted more often other type of performance testing, Stress Testing (checks behaviour of the software over and above anticipated peak load conditions - purpose is to find the upper limit and failure point), configuration testing (conducted to increase software performance by tweaking software or hardware configuration), volume testing (unlike load testing inputs very few concurrent users but increases the volume of data), Soak Testing (test over longer period time under continuous load conditions - can identify memory leaks, network and database connections leaks) and Spike Testing (conducted to find ability to withstand sudden increase in load - for example morning and evening login actions in gmail, yahoo, facebook).

Performance test planning is very important part of the performance testing. Well prepared performance test plan includes requirements (application flows, VU users, response times, transactions to be monitored, testing scenarios, tools to be used, load generators, resources, documentation and estimate the load test efforts). The other actions what are needed to run the successful performance testing are prepare the testing (create VU (virtual user) test scripts for workflows and define the test data), create scenarios (that describe the events during the testing sessions and details on machines), execute the scenarios, monitor the scenarios and analyze the results and do the conclusions and needed fixes, repeat when needed.

Performance analysis (or profiling tools) that can help to highlight hot spots and resource usage

Types of Performance Testing

1. Load Testing - most often done performance test.

Objective of load Testing is to check the behaviour of the software at normal and anticipated peak load conditions, Load testing is most often done with automated tools. Load Testing is performed to check if critical functions are working as expected, software response time and throughputs are at acceptable level. Load testing is simulating multiple concurrent users in the system.

2. Stress Testing

Objective is to find the upper limit or failure point where the software crashes. By this can be proved that software won't crash and does not have data loss or data corruption at peak loads. In stress test is conducted to check the software behaviour over and above the anticipated peak loads.

3. Soak testing, also called Endurance or longitivity testing

Software is tested under continues load conditions where memory utilization is monitored, throughput and response times are checked after some, long time. In this tests memory leaks, network and database connection leaks can be identified.

4. Configuration testing

In configuration testing is intended to increase the software performance by tweaking software and hardware configuration.

5. Spike Testing

Spike Testing is conducted to find ability of software to withstand in a sudden load increase (popular web-sides like facebook). Spikes are generated when most of the user for example do login or check email transactions.

There are many performance tools available in the market, both payable and open source tools. Most commonly used performance testing tools are for example LoadRunner, IBM Rational Performance Tester (RFT), and opens source tools like LoadUI, OpenSTA, jMeter.

During the performance testing following performance testing metrics are captured: TPS Transactions per second, average response times, throughput, peak response times, error rates, requests per seconds and concurrent users.

Performance testing parameters measured during the performance testing: physical memory usage, virtual memory usage, CPU utilization, and CPU interrupts per second, disk usage, database connections, database locks, network usage, network bottle-necks, and hits per seconds, bandwidth usage, number of active sessions and number of open connections.

Performance test planning follows the normal test planning procedures from requirements gathering through test case creation and test execution to analyzing the test results and needed corrective actions to be taken, (SoftwareTestingSoftware, 2014).

4.3.6 Security Testing

Security testing requirements can be classified for example:

Software Security, Data Security, Information Security and Network Security.

Software Security secures process, tools and deliverables during the software development life cycle (SDLC), Data security focuses to backup, encryption and masking the data. Data masking can be also used for test data when it is copied from the production environment. Data Security focuses for methods and policies to prevent for example the accidental damage. Information security focuses on protecting information for example from unauthorized access. Network security prefers to monitor and prevent unauthorized use for example hackers, worms and malicious software's.

In the company level there should be higher level security policy established, security risk and mitigation plan, proposal for adopting of industry security standards, training plan about policies and security measures defined.

Security testing is needed because some applications are at high risk for example finance and military domains and web-applications. Some of the reasons why security

testing is so important are: DownTime - revenue loss always for the company when service needs to keep down and unavailable for the customers, Legal Issues - Security Issues can lead to the legal issues, for example SSN numbers and bank accounts are hacked and published in web, Brand Damage - Continues DownTimes or hacking issues are easily causing damage for the Image and Brand and Costs - sooner the security issues are identified in a development process less costly issues they are.

Software testing Software is introducing seven attributes for security testing:

Authentication - identification of a person or an user for example by UserID and password, by bank security identification, token,

Authorization - when user is authenticated access is limited,

Confidentiality - system's ability to protect the information or data from Un-authorized use (one example of confidentiality is SSN stored in database should be encrypted),

Availability - refers to accessibility of the information system,

Non-repudiation - confirmation send by receiver to sender

Resilience - refers to Resistance of attacks, can be built 2 layer authentication, one time password, encryption, SSL.

Security testing process is from process perspective quite identical compared software testing process but couple of thing is good to high light because of the importance.

Traceability matrix, document security test and the end report what is including detailing risks, vulnerabilities, threats and open risks if any.

There are type of tests what can be performed during the security tests, they are:

SQL Injection, port tests (open ports - starting point for vulnerability scan), Vulnerability Scan (port scanner, Network, Database and web application vulnerability scanners),

Server Validation (client and server side validations, for example if web application is validating only using JavaScript it creates security vulnerability), session management

(session ids random and alphanumeric string), Browser Caching (not to store Bank

account or SSN numbers), Database Policies (confidential data encrypted or masked).

Ten questions to raise for your potential cloud provider to ensure your data is secure as identified, (Synccplicity, 2014).

1. What is your data encryption philosophy, and how, especially you encrypt data?
2. How do you manage the encryption keys?
3. What certifications for data protection you have attained?

4. How much data replication are enough and what level of data durability do you provide?
5. How much control do I retain over my data?
6. How do you ensure your client (endpoint) security?
7. Can I leverage existing credentials and password policies and disable access immediately?
8. How do you isolate and safeguard my data from that of other clients?
9. How is activity in my account monitored and documented?
10. Can you continue to provide as my workloads evolve?

4.3.7 Software Delivery

Software Testing is one part of the activities what are done before Software delivery what needs to be done to make a system available for use. Delivery is a general process consists of several activities and possible transitions between activities and what has to be customized based on the requirements.

Delivery activities are release (activity follows the development process), install and activate (make all the supporting systems ready to use), deactivate (inverse of activation, is needed for new deployments), adapt (modifying the system, for example changing the customer side environment), update (replaces the earlier version), Built-in (mechanism for installing the updates), version tracking (helps the user to find and install updates), uninstall (removal of the system/software), retire (end of the lifecycle of the product), (SoftwareDelivery, 2014).

Delivery roles are essential for successful delivery. The roles are specialized for coordinating and engineering the process. Roles are involved change an application progresses from testing environments to the production environment. Application developers, Build and Release Engineers, Release Managers and Deployment Coordinators all have important role in delivery/ deployment process, (StackExchange, 2014).

4.4 Alternative Software Delivery Concepts

4.4.1 "WaterScrumFall" - What is that?

There can be identified some challenges in agile planning for example top and middle management needs vision, roadmap, schedule and plan. In the company level there are strategy and budget and for those the high level planning is based on. But agile

team thinks that is not Agile. This is one of the root causes why some companies create a Hybrid of Agile and Traditional project Management.

Traditional Project Management is used to clarify the front plans. When that is done, agile team takes over, the team will follow the agile practises in product development and release process. In practise two different approaches are applied, top management follows the tradition way and thinking when leading planning the company activities and IT-development follows the Agile approach in development and release.

To get this hybrid process working different process philosophies, methods, terms and metrics to be created and applied. The message in the whitepaper is that product roadmap milestones and developed market releases must be aligned completely with software releases and Agile sprints.

Full traceability matrix is needed in Agile Practices from plans to deliverables by this can be sure that there is a tracking from the requirements to deliveries and how it reflects to the roadmap. If something is not completed that is visible. Requirements should have priorities based on what is the most important for the customers and market dynamics.

In the article there was the fifth challenge of agile projects - Losing the Forest for the Trees. If the Agile project team faces the situation where the backlog is increasing, new things are coming into the mix, more bugs than fix and people in the project become tired, the project is going to lose the scope and schedule. There is a fair that team start to deliver less valuable features, small things instead of the main thing because it is more complex and challenging and requires full effort and concentration from the whole team. If that is already happening in the project Product Owner should whistle the break in the game and lead the whole team back to basics - clarify the vision and listen to the customer input - the project team should out all the focus to the main thing and understand really what it is.

Product Owner is in the position where he/she can and must do prioritizing; the most valuable things to the customers and to the market is in top of the list and will be delivered at first and less important and not critical bugs to be ignored or fixed and delivered on later iteration phases.

Summarizing some of the main things Agile projects should be starting from the clear vision and what you would like to solve for the customers. Clarify who is the customer; what is the value in the system or in the scope. Make the fast customer feedback loop that will guide the design and priorities and validate your product or solution. Communicate agile methods, metrics and terms to the business leaders. When necessary refocus for the most valuable features, (Jama Software, 2011) .

4.4.2 IBM introducing Global Delivery Excellence Best Practices

There are many challenges in organizations from delivery perspective. One of those is adapting Global Delivery Excellence Best Practices for improving software process and tools adoption. Most of the organizations want to speed up the deliveries, improve the software quality and integrate across silos and simplify the complexity of the deliveries.

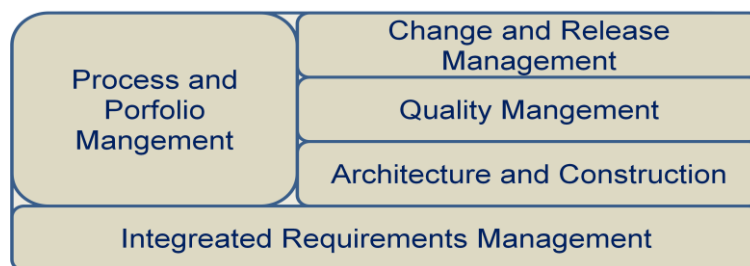


Figure 11: Global Delivery Excellence Best Practices (IBM, Shah, 2008).

IBM, Shah (2008) recommends supporting efficiency, control and value at every level with four core areas:

Software and System Delivery Solutions, simplifying SOA lifecycle management with reusable methods and assets.

Governance and Lifecycle management, implementing the best practices, processes and expertise to get more visibility and clarity across projects and processes.

Collaboration and global software skills, community based collaboration for the global teams and distributed education and infrastructure.

Productivity and automation by implementing the best practises and products to improve development and delivery processes.

IBM's approach for delivery excellence are to adopt the set of common tools, methods and processes iteratively, employ reuse of assets, training of development tools, demonstrate quick-wins in projects and implementing a collaborative software delivery

platform. According to IBM Software Universe four Pattern for the success are Scope Management, Process Management, Progress Management and Quality Management.

4.4.3 Two kind of delivery models

Clients can choose what type of delivery model for the software delivery they would like to get. Comparison of two kind s of delivery models for software products and services, Appendix 5: Two Software Delivery Models.

The first thing to be analyzed in the organization is the Delivery Model what to be selected for the software solution. There are two kinds of delivery models for software products and services, on-premises software and software as a Service (SaaS).

In-premises software has many names such as desktop software, enterprise software and shrink-wrap software. User can access to the on-premises software from the computer either locally (desktop programs) or via desktop that is connected to the internal network of the organization. Support is organized and hosted locally.

SaaS model is sometimes referred to as on-demand software, cloud-based software or web-based software. Hosting of SaaS software is done by the vendor and access happens via the Internet. Examples of cloud based software are on-line purchase web-shops, ticket reservation software. User is accessing the transaction-based applications via Internet and storing the data on the desktop or to network, (StaffReady, 2014).

4.4.4 Continues Delivery

Continues Delivery model is the last step in a long process. It might take a year to start making several deployments a day. The commitment from the management is needed for the process. Continues Delivery model will change the whole development methodologies and affect everyone in R&D.

The whole development methodology will be changed to Test Driven Development. The key concept of Continues Delivery is that everything should be tested automatically. One reason for long release processes is manual testing and regression testing takes also long time, (Mordo, 2013).



The Deployment Pipeline is a foundation in a process of continuous integrations. Almost all the testing should be automated but there are some things cannot be automated or are impossible to automate. Exploratory Testing relies on experienced testers. The list of things that cannot be automated is much smaller than many people think, (Humble, Farley, 2011, p.4,25).

Build Quality In means that that testing is not a phase. It starts from the development. Secondly Build Quality In means that testing is the domain of testers, the whole delivery team is responsible of the quality, (Humble, Farley, 2011, p.27).

The most common external dependencies within the application are the third party libraries it uses and relationship between the components and modules under deployment, (Humble, Farley, 2011, p.38).

The key to managing environments is to make their creation a fully automated process, (Humble, Farley, 2011, p.50).

Automated Acceptance testing have a number of valuable properties, for example make the feedback loop faster, reduce the workload on testers, free testers can concentrate exploratory type of testing, regression testing is automated (advantage when developing large applications).

Once automated acceptance tests are in place business acceptance criteria of your application is tested, validation for the valuable functionalities for the users are provided. (Humble, Farley, 2011, p.86, 187).

4.5 Conceptual Framework

The conceptual framework is created based on the selected IT project related areas and searched alternative software delivery concepts. The selected tools/areas under examine are IT project facts, Project Management, Software Testing (framework) and Alternative SDC - Software Delivery Concepts.

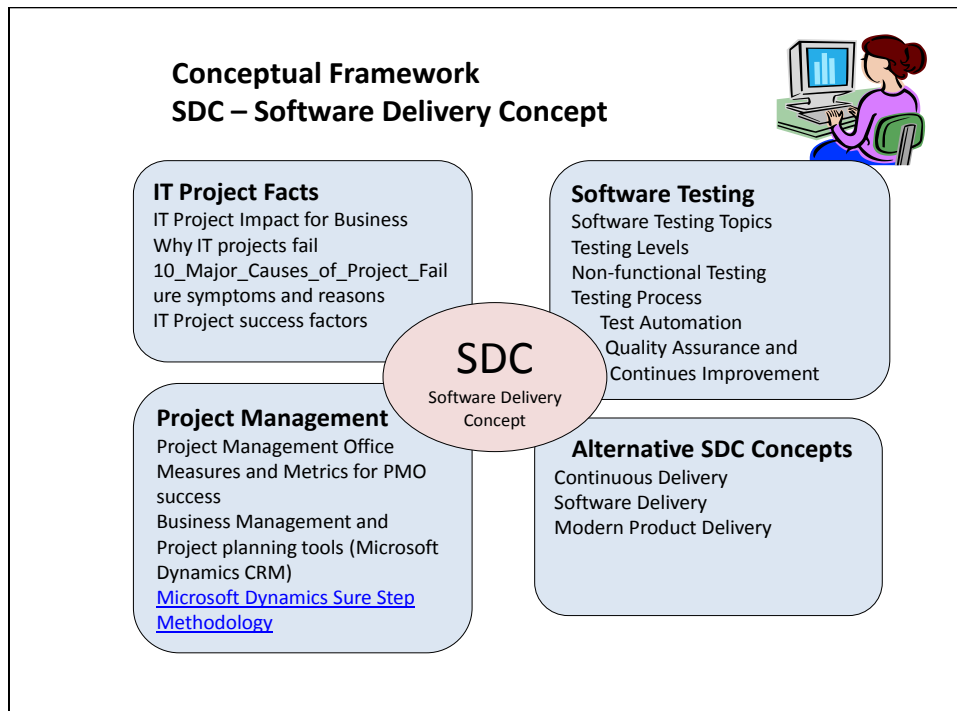


Figure 12. Conceptual Framework: SDC - Software Delivery Process

Software Delivery Concept (SDC) and how it has been examined in this project has been divided for different main areas. Those main areas are IT Project Facts, Project Management, Software Testing and Alternative Software Delivery Concepts.

For each of the areas are selected the main topics to cover.

IT Project Facts area is covering few articles and information how IT project impact for the business, why IT projects fail, 10 Major causes why projects fail - symptoms and reasons and IT Project success factors.

When looking for the root causes for project failures can find more commonly prominent issues what are causing delays or more serious impact for the progress of the project. List of the ten major root causes for the project failures is already in 2004 done list of items causing failures. The list is still very valid and showing the main topics quite

common for many failure projects. From the different sources found topics are much aligned with this list but the priority of the order can be different.

10 Major Root Causes for the project failures are:

1. Change Management, 2. Poor Communication, 3. Insufficient / inadequate Resources, 4. Poorly defined requirements, 5. Inaccurate estimates, 6. Poor Risk Management, 7. Poorly Defined Deliverables, 8. Over Optimism, 9. No time for Project Management (including roadmap and controlling) and 10. Improved Project Management skill set is needed, (Centreline Solutions Inc., 2005; V02.18)..

Prioritization of projects, utilization of project portfolio techniques and continuous improvement are the key things to follow for successful implementation.

Both business and IT are to blame for project failure. Successful IT projects organizations do need establish a project management framework with four core elements like processes, organizational structure, people and systems and a effective project management methodology is required for managing scope, time, resources, change, risk, cost, issues, configuration, quality and communication.

Many times in the projects is heart about **poor communications between business and IT**. One solution for poor communication between IT and business is that IT people really need to try to understand what the business requirements are and what business people are telling them. IT people are in key roles in the project.

Unclear Business Requirement makes unclear expectations

In order to avoid ambiguous expectations in a project it is necessary to understand and fully outline requirements and communicate clearly for stakeholders at the start of the project. Projects produce the solutions what are based on business requirements.

Critical Project Success factors are influenced from four different areas what are customers (customer requirements and expectations), processes (standardized and controlled), people (stakeholders and project teams: leadership and management, teams) and technology (IT systems, workflows and tools).

Project Management - Best Practices

Wiegers (2014) introduces valuable practises which are organized into five categories.

1. Laying the foundation

Stakeholders should have a common understanding about project success, business interest and expectations. The project should also have clear and measurable goals.

2. Planning the project

The time what is spent analyzing what it will take a solve the problem will reduce the number of surprises in the project. Useful plan includes lots of things to be considered: resources, budget and estimated plans for those. Teams are distinctly defined and responsibilities are determined.

3. Estimating the work

Wiegers (2014) prefer to estimate the effort in labour-hours associated to the specific task and then translating the effort into the calendar-time estimate.

4. Tracking your progress - Record actual and estimates

Each individual should record the estimates and actual and project manager/lead should track this project data items on a project task or milestone basis.

5. Learning for the future

Retrospectives (post-project reviews) will help the project team to develop the process improving the done activities.

The advice is that every time when initiating a new project builds the corresponding activities for the project. Those activities which add value for the project, only those activities should be applied.

Project Manager's Agile Survival guide by Rally Software

Rally Software (2014) has published an interesting article about Project Manager's Agile Survival guide. The main message in the whitepaper is that agile project and so called traditional project are not that far from each other's and all the roles and practices can be mapped for corresponding roles and tasks.

Agile refers to practices that encourage rapid deployment of working products, focus on quality and value and integrated cycles of feedback.

Mapping of Waterfall Project Management to Agile Practices

Rally Software (2014) depicts that mapping of Waterfall Project Management to Agile Practices is not the huge task and Agile Practices can help on managing the complex

work and improve project delivery, agile projects are iterative and incremental, feedback loop is fast. Agile and Waterfall development aren't as different as people image they are. There are differences in these two approaches in the implementing costs, schedule and scope what both are recognizing.

Agile project phases map nicely to iterative development environment. This means in practice scaling from short iterations up to longer-term releases.

In different organizations agile way of working can be adopted at a speed that works best for them and agile practices can be customized. Utilizing the agile in portfolio level changes is needed in organizational level, for example in processes, structure, roles, reports and metrics.

"Subway map" by Agile Alliance

"Subway map" illustrates very logical way the different areas of concerns (Extreme Programming, Teams, Lean, Scrum, Product Management, Design, Testing, Fundamentals), Agile Alliance (2013).

Five challenges to Agile Planning

Jama Software has released an interesting article about five challenges to Agile Planning. Author of the article is Dorian Simpson, Managing Partner, and Planning Innovations.

Even most of the Agile projects developers would like to start the works immediately there must be given sufficient time to provide the business requirements what is the basic for the product and for the project. Business requirements in higher level give the directions for the projects and also for the development teams what is expected from them.

There is a need to create a vision, product plan and prioritizing the use cases, preliminary design, market analysis and so forth. Development team is involved for architecture, platform and performance planning.

Product Owner has a leading role in orchestra, the decision-maker.

Software Testing

Pre-conditions for software testing - Entry Criteria

Before it is reasonable to start testing there are some pre-conditions to consider.

Without taking any deeper sight to the every topic in the entry criteria, the main topics are listed: Test environment (set up for environment, configurations, interfaces), roles and accounts are defined (correct roles with access rights), test planning is done (test strategy, test plan, test scenarios/cases), previous tests are successfully done, no open critical issues and test data is defined/available.

Software testing in general

Software Testing is a set of activities what is performed to verify and validate the software against the requirements. Is the software built right (according to the specifications) and is the right software built (according to the customer requirements).

Software Testing Life Cycle (STLC) model is identifying what Testing activities need to be executed and what ii the best time to perform the activities. Before Software Testing Process can be started Requirement Analysis should be done.

Software testing can be seen as a part of Software Testing Life Cycle. There are different testing approaches like manual and automated testing and non-functional requirement testing. All these approaches can be included in a testing process.

Typically test cycle includes test execution and bug reporting. Test results should meet the plans. There can be one or several test cycles. Before moving to the production final tests and the decision should be done to move to the production. Needed documentation/evidences about testing and test results to be available for the decision making. Roll-back plan and actions to be taken before moving the product to the production.

In post implementation phase production smoke test will be done. Lessons learned will be collected to avoid the faced issues in the future releases.

Manual Testing

Manual Testing is the process of executing software to validate against the requirements. In manual testing aim is to find the software bugs / defects before release in production. Manual testing is done by human being not by the machine / automated

tool or by script. There are different phases in manual testing for example unit, integration, system and user acceptance testing, operational readiness test, alpha and beta testing and smoke testing. Manual testing may involve various interfaces like GUI (Graphical User Interface), CLI (command line interface) and /or automated frameworks to test APIs.

Manual testing typically involves using Black-box, White-box and /or Grey-box testing methods depend what type of software testing is performed.

Manual testing can be performed as Statistic or dynamic testing. Statistic testing means testing software against the requirements, verification and dynamic testing refers to execution of software uncover defects / software bugs, (SoftwareTestingSoftware, 2014).

Regression Testing

In regression testing the focus is on finding defects after the code change is done. In regression testing main processes are retested and ensured that existing functionalities and working and old issues have not come back again. Efficient regression testing requires well-developed test suite and automation in testing. Automated functional GUI testing tools are used to repeat system-level tests through the GUI.

Regression Testing is initiated after programmer has attempted to fix a recognized problem or has added source code to a program. Newly added code should still comply with its specific requirements and unmodified code has not been affected. These can be measured by quality control measures.

At any time software is implementation modified regression testing is needed.

In Regression testing is tested that bug fixes are not impacted for other software functionality of the application and the software is working as it is expected to work before the code changes are done. Regression is generic and it is not necessary specific for any bug fix or code change or involve those changes but only execute the separate regression testing cases.

Efficient regression testing requires well-developed test suite and automation in testing. Automated functional GUI testing tools are used to repeat system-level tests through the GUI.

When changes against the existing functionalities are regression tested, new testing cases typically become as "regression testing cases and to be added in regression testing case library or existing regression cases to be modified including the newly introduced changes.

Building a Regression Test case Library

The most effective approach is to build up a regression test case library. Every time when there is a new build or change in the software regression tests (test battery) will be run from the library.

Automated testing is typically required

Guidance is to spend as less as possible time with manual regression testing, Regression testing should be automated.

There are various automated testing programs that make regression testing relatively easy to manage, couple of clicks of a mouse to establish set of testing parameters and to check new iterations of code against previous software version. Inconsistencies are checked from the logs and specified where the code broke and why.

In most cases there isn't time to run the entire software through but coverage is dramatically better than doing testing manually.

Regression tests are seen effective and cost-efficient, comprehensive part of the testing strategy and test planning. Regression testing is an essential aspect of a dynamic, iterative development and deployment schedule. Well-planned testing strategy with automated regression testing at its core, project keeps the planned budget, team is kept on track, unexpected bugs are prevented to damage the project.

Performance Testing

Performance Testing is validating non-functional requirements (response time and throughput) and validate software quality (reliability, scalability and efficiency). Performance testing has become vital for example for web-applications. Those compo-

nents what affect for the performance goal (response time, throughput and scalability) are called performance bottlenecks. Objective of performance testing is to find the bottlenecks and to retest when performance fixes are done.

There can be many type of issues causing performance bottlenecks, some examples: frequent updates to log files or database tables, not enough hardware resources for running software, network performance issues, database is not indexed, database and SQL queries have performance issues.

Performance Testing Types are for example Load Testing (checks the behaviour of the software under normal and peak load conditions) - conducted more often other type of performance testing, Stress Testing (checks behaviour of the software over and above anticipated peak load conditions - purpose is to find the upper limit and failure point), configuration testing (conducted to increase software performance by tweaking software or hardware configuration), volume testing (unlike load testing inputs very few concurrent users but increases the volume of data), Soak Testing (test over longer period time under continuous load conditions - can identify memory leaks, network and database connections leaks) and Spike Testing (conducted to find ability to withstand sudden increase in load - for example morning and evening login actions in gmail, yahoo, facebook).

Performance Test Planning

Performance test planning follows the normal test planning procedures from requirements gathering through test case creation and test execution to analyzing the test results and needed corrective actions to be taken. Performance test planning is very important part of the performance testing. Well-prepared performance test plan includes requirements (application flows, VU users, response times, transactions to be monitored, testing scenarios, tools to be used, load generators, resources, documentation and estimate the load test efforts).

Types of Performance Testing

1. Load Testing - most often done performance test.

Objective of load Testing is to check the behaviour of the software at normal and anticipated peak load conditions, Load testing is most often done with automated tools. Load Testing is performed to check if critical functions are working as expected, software response time and throughputs are at acceptable level. Load testing is simulating multiple concurrent users in the system.

2. Stress Testing

Objective is to find the upper limit or failure point where the software crashes. By this can be proved that software won't crash and does not have data loss or data corruption at peak loads. In stress test is conducted to check the software behaviour over and above the anticipated peak loads.

3. Soak testing, also called Endurance or longitivity testing

Software is tested under continues load conditions where memory utilization is monitored, throughput and response times are checked after some, long time. In this tests memory leaks, network and database connection leaks can be identified.

4. Configuration testing

In configuration testing is intended to increase the software performance by tweaking software and hardware configuration.

5. Spike Testing

Spike Testing is conducted to find ability of software to withstand in a sudden load increase (popular web-sides like facebook). Spikes are generated when most of the user for example do login or check email transactions.

The other actions what are needed to run the successful performance testing are prepare the testing (create VU (virtual user) test scripts for workflows and define the test data), create scenarios (that describe the events during the testing sessions and details on machines), execute the scenarios, monitor the scenarios and analyze the results and do the conclusions and needed fixes, repeat when needed, , (SoftwareTesting-Software, 2014).

Security Testing

Security testing requirements can be classified for example:

Software Security, Data Security, Information Security and Network Security.

Software Security secures process, tools and deliverables during the software development life cycle (SDLC), Data security focuses to backup, encryption and masking the data. Data masking can be also used for test data when it is copied from the production environment. Data Security focuses for methods and policies to prevent for example the accidental damage. Information security focuses on protecting information for example from unauthorized access. Network security prefers to monitor and prevent unauthorized use for example hackers, worms and malicious software's.

In the company level there should be higher level security policy established, security risk and mitigation plan, proposal for adopting of industry security standards, training plan about policies and security measures defined.

Security testing is needed because some applications are at high risk for example finance and military domains and web-applications. Some of the reasons why security testing is so important are: DownTime - revenue loss always for the company when service needs to keep down and unavailable for the customers, Legal Issues - Security Issues can lead to the legal issues, for example SSN numbers and bank accounts are hacked and published in web, Brand Damage - Continuous DownTimes or hacking issues are easily causing damage for the Image and Brand and Costs - sooner the security issues are identified in a development process less costly issues they are.

Software Delivery

Software Testing is one part of the activities what are done before Software delivery what needs to be done to make a system available for use. Delivery is a general process consists of several activities and possible transitions between activities and what has to be customized based on the requirements.

Delivery activities are release (activity follows the development process), install and activate (make all the supporting systems ready to use), deactivate (inverse of activation, is needed for new deployments), adapt (modifying the system, for example changing the customer side environment), update (replaces the earlier version), Built-in (mechanism for installing the updates), version tracking (helps the user to find and install updates), uninstall (removal of the system/software), retire (end of the lifecycle of the product), (SoftwareDelivery, 2014).

Alternative Software Delivery Concepts

"WaterScrumFall" - What is that?

There can be identified some challenges in agile planning for example top and middle management needs vision, roadmap, schedule and plan. In the company level there are strategy and budget and for those the high level planning is based on. But agile team thinks that is not Agile. This is one of the root causes why some companies create a Hybrid of Agile and Traditional project Management.

Full traceability matrix is needed in Agile Practices from plans to deliverables by this can be sure that there is a tracking from the requirements to deliveries and how it reflects to the roadmap. If something is not completed that is visible. Requirements should have priorities based on what is the most important for the customers and market dynamics.

IBM introduces Global Delivery Excellence Best Practices for improving software process and tools adoption

IBM, Shah (2008) recommends supporting efficiency, control and value at every level with four core areas:

1. Software and System Delivery Solutions, simplifying SOA lifecycle management with reusable methods and assets.
2. Governance and Lifecycle management, implementing the best practices, processes and expertise to get more visibility and clarity across projects and processes.
3. Collaboration and global software skills, community based collaboration for the global teams and distributed education and infrastructure.

IBM's approach for delivery excellence are to adopt the set of common tools, methods and processes iteratively, employ reuse of assets, training of development tools, demonstrate quick-wins in projects and implementing a collaborative software delivery platform. According to IBM Software Universe four Patterns for the success are Scope Management, Process Management, Progress Management and Quality Management.

Continues Delivery

Continues Delivery model is the last step in a long process. It might take a year to start making several deployments a day. The commitment from the management is needed for the process. Continues Delivery model will change the whole development methodologies and affect everyone in R&D.

The whole development methodology will be changed to Test Driven Development. The key concept of Continues Delivery is that everything should be tested automatically. One reason for long release processes is manual testing and regression testing takes also long time, (Mordo, 2013).

Two kinds of Delivery models for software products and services

The first thing to be analyzed in the organization is the Delivery Model what to be selected for the software solution. There are two kinds of delivery models for software products and services, on-premises software and software as a Service (SaaS).

User can access to the on-premises software from the computer either locally (desktop programs) or via desktop that is connected to the internal network of the organization. Support is organized and hosted locally.

SaaS model is sometimes referred to as on-demand software, cloud-based software or web-based software. Hosting of SaaS software is done by the vendor and access happens via the Internet. Examples of cloud based software are on-line purchase web-shops, ticket reservation software. User is accessing the transaction-based applications via Internet and storing the data on the desktop or to network.

Clients can choose what type of delivery model for the software delivery they would like to get. Comparison of two kinds of delivery models for software products and services, (StaffReady, 2014).

Appendix 5: Two Software Delivery Models.

5 Creating a Prototype Concept

5.1 Creating a Prototype Concept and Theme Interviews

Both qualitative and quantitative research methodology was utilized in this study. The qualitative research data consists of six in-depth interviews. Two of the theme interviews were done with project people and four of them with people who are working in international companies in IT-projects. Quantitative data research data was gathered with the questionnaire. Six questionnaires were sent for IT-project professionals in couple of international companies, four responses were received.

Theme Interviews have been made for six people, two of them are from the project and four of them are IT-professionals outside the project. Theme interviews covered the areas about projects, project management and release and testing process related questions.

People who were interviewed are working in international companies and are all working in IT-projects and / or in process development projects in different positions. Five of

them are re-presenting managerial level and one is a lead developer in a team. People from the project are having the roles of product owner (project manager) and IT architect and people outside of the project all are IT professionals, three of them are either project/managers or one pre-present development view.

All of them are working in international companies and projects are global. People outside of the project are coming from well-known, big companies operating worldwide. Projects are global and some of them are having local requirements. In big programs delivery teams are local teams reporting to the corporation functions.

Appendix 6: Theme Interview Questions

Appendix 7-12. Theme Interview topics

5.1.1 Project Management

What are the key success factors (3 to 5 pieces) in your projects for on-time product / software delivery?

All the interviewed people said that two very important things for on-time product / software delivery are good communication within the project team and regular follow up for the activities. Depending on the role in a team activity follow up could vary; for project managers it could be regular daily, weekly or monthly meetings (onsite or online) to follow up, monitor project and activity and delivery progress, or when needed also more radical changes can be done for the approach, priorities and resources, for team members in development or testing team it can be daily, continues, open communication and knowledge and information sharing about the situation and activity progress. Person who work as a program manager raise one issue even more important than those two common things for all, it is stakeholder (business owners and users) involvement.

Project planning is essential for each of the successful project.

Key component for any project delivery is the quality of the resources. Right skill set of resources should be allocated and working on the project.

The project plan should be addressed for all the activities from the small detail to the dependencies listed. Effort for each of the activities should be identified. The critical path should be identified at the start of the project. Progress on the plan should be updated periodically to the respective stakeholders and also to the project governance teams.

Understanding of the Client Expectations/ Requirements: It is very important that the client requirements are understood both the explicit requirements and the implicit un-said ones. The requirements of the client can be identified, if the requirement gathering process concentrates on what is the business need and why the customer would require the change/ system implementation to be done.

Running the successful on-time project requires effective and accurate project planning: project plan, project schedule, resource planning, communication plan, plan for testing and training activities and risk matrix and mitigation plan. All the interviewed said in a bit different words that robust and structured review process for the deliverables is must through the whole project and retrospective in the end of the project for learning purposes. Learning from the mistakes through the project and not only in the end requires continues and clear practises how to do. Also saw essential that requirements and project scope is clear from the beginning of the project.

Collaborative teamwork is a key for success in all teams. Roles and responsibilities should be clearly defined and agreed.

In the development side continues Integration and continues release testing with automation is in an important role in all IT-projects also see very important that design checks are done at every stage if major changes need to be done to get the project ready on-time.

What are the things (3 to 5 pieces) causing delivery delays or confusion in the projects?

Common things what have been found through interviews what are causing delays in the projects are unclear requirements and frequent changes in a scope (scope creep). If there are situations where the change in scope is accepted without doing a thorough impact analysis it can cause a major reason for delay in the project. There could be situations that Requirement gathering has had deficiencies, not understood the client requirements or recording them incorrectly and it will cause the delays of getting the requirements signed off. Also constant change in requirements impacts the estimates and the project plans. These two things are often causing delays right from the beginning of the project. Also found the big amount of scope items challenging to manage.

From the schedule perspective unrealistic deadlines and budgets are given many times.

In an international environment in global projects challenges are faced more often in the projects which are delivered for across functional organizations. Multi vendor environment in different locations cause delays in communication because of time differences. In some cases change management and communication especially in change management could be done better.

Unsigned documents have caused delays on starting the development work.

When the project has come into the development phase there has been some issues causing delays and confusion in development phase.

Planning the things properly in each phase is half work done. In development side if no proper initial estimation or elaborate planning is done it causes delays. In many cases things are at first done as a simple prototype to get the things working and only then the design plan is done. This is causing lots of rework. To avoid lots of rework prior planning and estimation should done.

In the example projects User Interface changes are accepted in the middle of the project or at any other stage of the project. When development and UI designing go parallel UI changes cause also possible frequent changes also in development side.

In some of the projects have been faced development discipline which means that in the project has not been followed the development discipline - in terms of coding standards, writing reusable code and comments in the code making it difficult for development and review. Also incomplete Testing is caused delays in the projects; not done a thorough testing of the code and also the impacted modules resulting in the delivery are being rejected.

The scrum process is followed only for planning and some developers are rather doing their individual work and task than working together and work as a team.

Detailed level task lists are not planned in sprint planning meeting. Only high level tasks are defined as a task. This is done because of unknown technical details as this stage. When technical details are know there isn't process and feedback loop defined to update and re-evaluate the sprint plan / task list and what things changes are caus-

ing. Sprint planning meetings are kept very short and in high level. This practise is causing risks that unexpected things are coming up in the middle of the sprint and then can cause delays for the sprint and for the whole project.

In testing phase frequent application down times are causing delays in testing. In some cases bug/defect fixes take more time than according to the agreed SLAs.

How long time is the release cycle in your process/project (from development to release in the production)?

Release cycles depend on the followed project approach. If the project is following waterfall approach release cycle can vary between 1 to 6 months based on the scope and if the project is following agile approach release process is from 2 to 4 weeks and amount of releases depends on a scope.

In some of the example projects release cycle in terms of development to release into production depends on the client environments and testing in UAT region. For a complex change that involves more than 100 MD of effort, the typical cycle time from development to Production would be 4 weeks. For medium complex change less than 100MD and greater than 20MD, release cycle would be 2 -3 weeks and for small changes it will be 2 weeks.

How do you prioritize the requirements? Who do it and based on what priorities?

In an example projects requirements are prioritized in a similar way; based on the business needs and based on what features would give the biggest value for product and for the business. Benchmarking other corresponding products in the market influences also for the prioritization of the requirement or the feature under development. IT team is responsible for communicating the requirements and priorities for the suppliers. IT teams prioritize the requirements keeping in mind the impact of these changes on related modules and the changes needed for these modules.

How do you analyze the dependencies between requirements and development items?

Development work starts only after requirements analysis and prioritization is done. Unclear or very complex requirements can in some cases deferred to the next release or development cycle. Sufficient requirement analysis will cover dependency analysis. By this practise bigger surprises can be avoided and possible delays in development work and in the project can be prevented.

In some of the projects requirement analysis was done very systematically covering mapping all the requirements into requirement tracking and tracking that the same are addressed in design, coding and in testing. Impact analysis for all the requirements were done, tracked the development and design addresses into the impacted modules. Testing should address and cover all the requirements and track the execution of the testing cases.

There are also example projects and releases in which requirement analysis has not covered dependency analysis and some scope items have been dropped out in the later phase. Often it works fine but can cause also lots of re-work in development side and extra work and additional testing cycles in testing side.

How costs are followed:

a) at project portfolio level and periodically

b) at project level and monthly bases?

c) more detailed level and more often? Explain how.

Most of the example projects the project costs are estimated only in project level, mostly monthly. Project cost estimations are based on the estimation of the costs in different areas in a project (costs of the work done in a different teams -labour costs (team size and salary costs), infrastructure costs, and additional ad-hoc costs). Cost estimations from the different area are calculated together. Composition of cost estimates is project cost estimation.

Similar way it is handled in a program and portfolio level. Costs are followed and reported only on monthly bases or within the agreed period. Additional costs can be caused for the programs and projects from the resource pools.

When some of the people are allocated for the project and if not necessary utilized for the work with 100% allocations are booked to an actual costs. In those cases always projects try to rotate people between the different tasks but in all cases it is not possible, for example if competences are not fulfilling the requirements. Mostly in development side people are responsible to update their knowledge base. If the demand of development work is bigger that can supply from own resource pool then pool from subcontracts are used.

Think of your last project. What are the things (3 to 5 pieces) could be done better?

There has been found some common thing in the project what could be done better in the projects like for example requirements analysis and prioritization of the requirements. Importance of requirement analysis come forth especially in the large projects where amount of scope items is very big. The aspect of the business risk has been quite weak in many cases in the requirement analysis. Risks should be clearly listed and categorized. There are some examples how the risks can be categorized, for example based on risks can cause the delays in the project, based on the technology can cause unexpected issues in the project and for example based on the business impact and innovation (new software introduced for the business users) what can cause unexpected delays for the project schedule.

In some projects is learned that implementing the process improvement methods like six sigma and Lean methods can be done better and more efficiently.

As stated earlier in an international environment in global projects challenges are faced more often in the projects which are delivered for across functional organizations. In some of the projects rotating the resources from one project or module to another could be done better for example by efficient cross functional training.

Multi vendor environment in different locations cause delays in communication because of time differences. Communication and change management is often a challenge and can be improved. Analyzed lessons learned results in details could help to improve the communication within the team and reduce the gaps. When the areas where has been lack of communication or it has been poor are known it can be improved in the next project or release. Better change management is achieved in a same way when clear change management process is defined.

5.1.2 Release process

What are the five key things to speed up the release life cycle?

Interviewed people have realized few things to speed up the release life cycle. Two of the things were common for the all the projects test automation or usage of any tool to improve the productivity . In development side it means continues integration builds and for example testing it can mean utilizing test automation for regression testing, system integration testing and even partly for user acceptance testing.

The second common topic what came up is review practices in different part of the release process. Efficient review process and practises are needed to follow up carefully deliverables or the progress of the process step. In release process it can be in development side sprint reviews and in testing side it can be the daily testing progress - wrap up meeting. In team level it can be a team meeting to follow up the activities within the team.

Interviewed people also mentioned that proper coordination among multiple teams is important. Having a clear picture what is happening in business side, development team, release management team and in testing team helps on coordination work and make it easier to do the needed actions or changes immediately when those are needed. Avoiding frequent changes in one or two areas or changes in requirements helps to avoid generating additional work and re-work. Efficient requirement analysis and change management process helps to allocate the changes either to same release or to include the requested changes to the next release.

Speeding up the release process is two important things also came along in some of the projects. These two things were clearly defined requirements and deploying the automation for the different tasks and phases in the release process.

In development side is seen couple of additional things to speed up the process / progress. The first of these things are that one dedicated tester or test lead checks the daily build or one stable build in a week and gives list as feedback and major issues for people who are fixing the found issues. The secondly performance testing (non-functional testing) is done before certification testing. And third thing is to follow Alpha Beta releases and to do the best to get the final release go same way and same speed as Alpha Beta releases.

Is there a same release process for new development works, configuration changes and database changes?

Through the interviews came clearly forth that in most of the example projects release process is same for all different kind of changes however the timelines for implementing the changes can vary but it is dependent on the project. In some of the projects bug fixes can follow the faster and shorter implementation process than other above mentioned changes. Also in some of the projects if a development project requires configuration changes and database changes then the same release process is followed but in

case changes are only configuration or only a database modification, then a different release process is followed.

However in many projects faster and shorter process is not followed in implementation when done small changes or changes what have no dependencies for any changes or other development items. In many cases also much manual work is included; configurations are done manually in each of the testing environments.

Are all different kind of changes clubbed together or different approaches and different life cycles? If any of the previous processes differs from each others, explain how. How the planning is done?

In project level different approaches, like agile, and timelines are followed and project plans based on the scope of the project. However in implementation and in release process this type of analyzing the change and categorizing them for group of different kind of changes is not so often used or used not at all.

When to use incremental approach or always? Is there any difference in front-end and back-end development?

For this topic I got quite different explanations mainly because of different type of the example projects. Some were saying that for this topic they have no visibility how it really is organized because development work and testing activities were out-sourced for different vendors. In some projects web development took longer time than back-end development work because much effort has put to the screen designing (how the screen looks like, how it is to navigate, what intuitions users get and so forth).

Then in another type of the projects main releases were including most of the features and incremental approach applies for the minor changes and updates only.

Also in some of the projects incremental approach was only used for database development and for front end it was a complete release.

5.1.3 Testing

What tools are used for test management tasks, for issue management, for automation, for documentation?

In big global companies in the example projects Quality Center (QC) was used for Test Management, QTP and Selenium for test automation and Share Point and in one case

Documentum was used for document management in procurement side. In some projects there were dedicated teams working on the test support tools. These teams have created and setup labs for all test support activities (build setup, dashboards).

In a smaller projects and smaller companies is more common to use cheaper tools or open source tools and for example JIRA, Bugzilla and also Microsoft Sharepoint was used for issue management, Confluence for document management. There is also some plug-ins for JIRA to manage the test management in a same tool, also open source tool Selenium used for test automation.

How big part of testing is done with automated scripts in development and in further testing phases? (percentage of automation)?

Are there any examples of automated acceptance testing or partly automated acceptance testing?

Automation scripts are mainly used in the development side with Continues Integration and in Unit Testing. Developers commit the done changes in Continues Integration tool, the new build is created every time automatically. In the example project there was two build systems: one build system for major releases and the other one for the minor changes, improvements and for continues daily work. In development side exact percentage how many percentage of the tests are automated was not mention but can say that it is very high compared for the later testing phases, the later testing phases automation is not fully utilized.

There are examples where QTP-tool is used in back-end side and for example Selenium is used in front-end side. In the example projects there were no automation used in full end-to-end- testing. Some weeks after one of the interviews I heard from the Program Manager that in their case vendor has started to automate regression testing and more than 80 % of the test cases can be automated.

In the big projects were multiple parallel releases were implemented at the same time and the software was found stable, only on those cases were recommended to use the automation as a part of System Integration Testing and in User Acceptance Testing.

What type of security testing is done in your last project? Who did it and who analyzed the coverage?

In many projects security testing is not done at all. In some projects only authentication was tested. In some example projects there was a certification team doing security testing manually but coverage and scope was not clearly known. Automation was not used in security testing.

Then there were examples where security testing was done in the initial deployment of a version. Vulnerability check was done by external parties and the list is defined in the product charter.

What type of performance testing is done in your last project? Who did it and who analyzed the coverage?

Performance tests done were volume and load testing by performance testing team by performance test lead/engineering. Surprisingly in some big projects used amount of concurrent users were quite low, just some 250 users. Tests are planned and are done based on non-functional requirements. Review, analyze and approval is done by project manager and business representative.

There was also example projects where performance testing is done for every version release. External agencies check the volume of transactions, load testing, load throughput and also concurrency throughput.

How much automation used?

In performance testing full automation is utilized. In most of the projects full 100 % automation and in some teams dedicated performance testing team have done proof of concept to automate all the test cases. The outcome was that 90 % of the testing cases could be automated.

How regression testing is done, manually or automated?

Regression testing is done only in one project example utilization of full automated and in one project having automation and manual approach together. In all the other projects it is done manually but was changed in one of the projects after couple of months following the automation approach.

Is manual testing risk or opportunity? Why?

Manual testing is an opportunity as new scenarios are not included in the regression test suite and manual testing helps us to test the new scenarios. Also variants of the

test case scenarios can be tested using the manual testing mode. Manual testing also provides quick turnaround time for developers to fix and check the resulting fix is working or not. But on the other hand manual testing can be also a big risk if all the testing is executed manually. Testing coverage is less than what can be covered in automation and humans do more errors than a robot or automation.

If testing is out-sourced how costs are followed?

a) in a project level or

b) separate cost management for testing activities?

Testing activities and costs related to the testing are followed only in the project level.

5.2 Delivery Concept

5.2.1 Project Framework

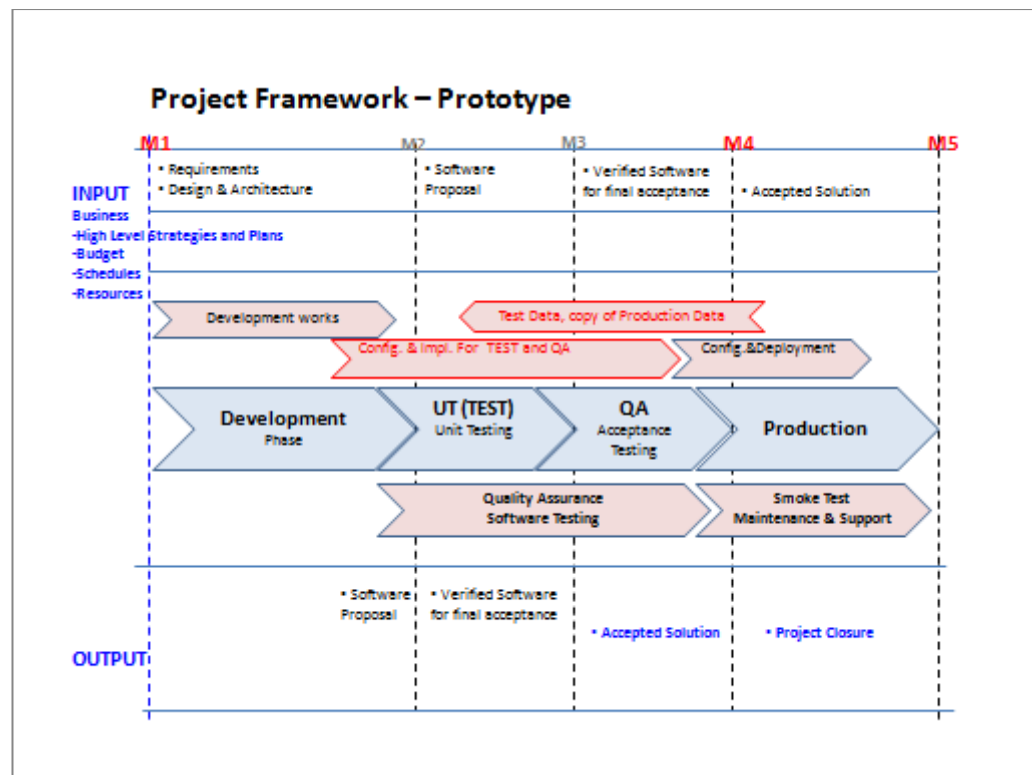


Figure 13. Project Framework - prototype

In the project Framework important milestones are M1, M4 and M5.

M1 started the actual development work. Business Requirement gathering is done and at least initially sign off by responsible people however scope creep had happened many times in a previous releases. Additional effort is required for the added or adjusted scope items. Practical guidelines have not official agreed with the vendor.

M2 and M3 milestones are kept as a check points how development works are proceed and how initial testing is proceeding. In the project or Release process those milestones do not require additional arrangements and development works can be continued until the official acceptance testing is done in QA environment. Software under acceptance testing is official Software proposal what is delivered for the testing environments (TEST and QA) at the same time.

Test Data for the testing is copied for the first time from the production environment and scramble.

Quality Assurance and software testing activities will continue until final acceptance is done (M4 milestone).

The product is ready for production deployment activities. Before the official production usage starts the smoke test of the delivered product will be executed successfully in production environment.

Before closing the project or release in M5 milestone lessons learned will be collected in post retrospective / lessons learned session and analysed. Analyzed results will be used to improve the way of working in a next release.

5.2.2 Development and Release Process - Back-end system

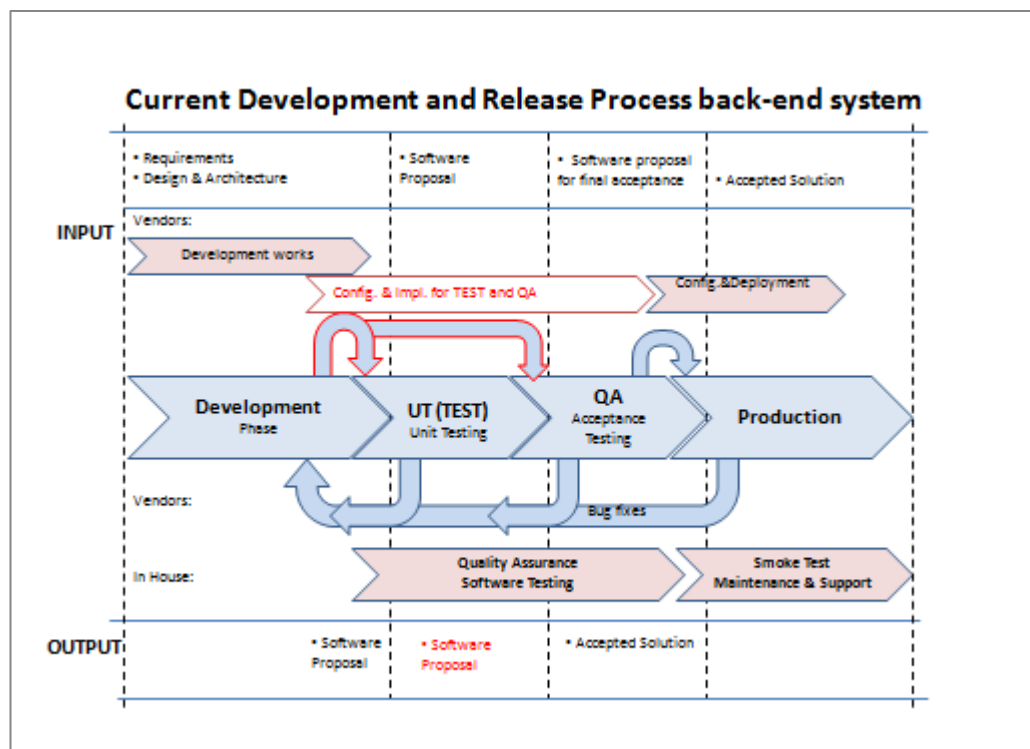


Figure 14. Proposed Development and Release Process - prototype

Release management is the process of managing software releases from development stage to software release in production. Currently there are three different phases before software release is ready to move to production environment for the real production usage of end-user.

With a current practise vendor do software verification in an own testing environment before it is moved for the testing in client side. Mostly the deployments are done separately for the testing environments (TEST and QA) and in prototype / pilot it product delivery will be done at the same time for the both testing environments.

Business requirements are continuously collected mainly based on the feedback from the end user and from the customer feedback. High level business requirements are coming from the management. Process is continues activity and prioritizing is coming based on the urgency.

All the development works (new development, bug fixes and configuration changes) are combined for the same release.

Co-operation with the vendor should be started in a deeper level. Sharing not only the testing results but also automated test cases would give the benefits for the whole release process.

In test management planning take Test Management tool into use to document test cases and testing result in a same place. Test planning boards in Test Management tool will be taken into use and available reports is tested to get the understanding how to follow up testing progress and give the needed visibility for the product owner and stakeholder, also in vendor side.

Test Management tool allows measuring the spend time on testing actives.

Almost all the testing is done manually except pricing checks and push purchase model from web without automated checking in backend software.

6 Testing a Prototype Concept

6.1 Concept Proposal: Software Delivery Concept

6.1.1 Proposed Improved Project Framework

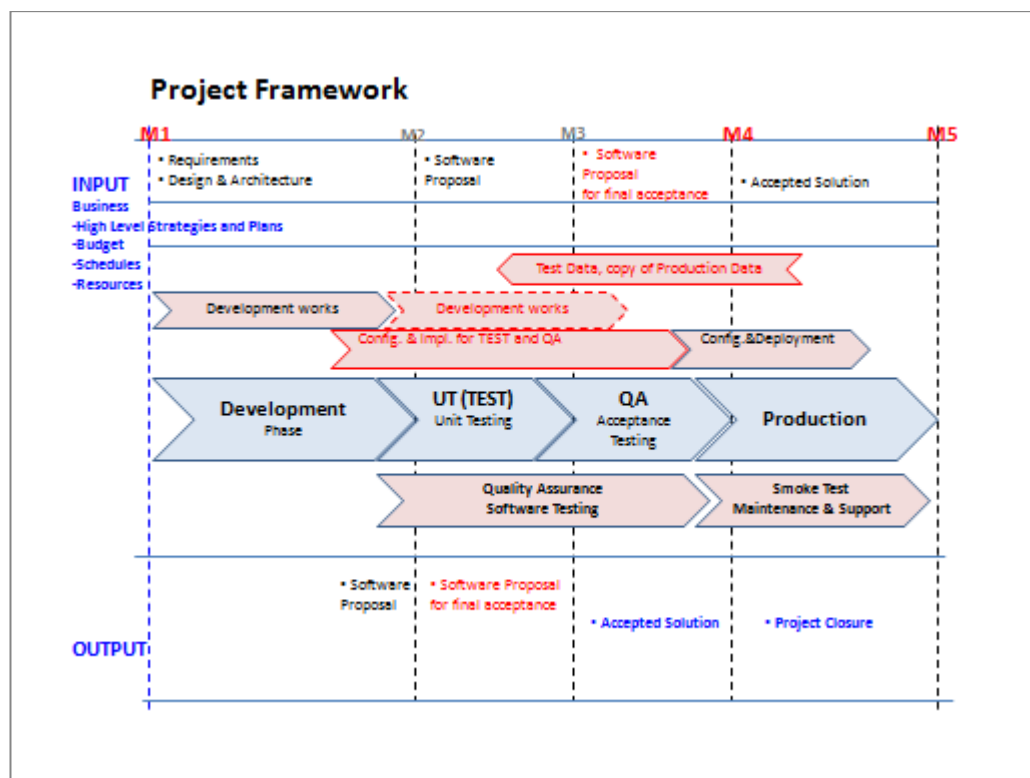


Figure 15. Proposed Improved Project Framework

In the project Framework important milestones are M1, M4 and M5 those will remain as they are in prototype model. M1 starts the actual development work. Business Requirement gathering is done and at least initially sign off by responsible people. Scope creep should be avoided. It will cause additional work for all but on the other hand when requirements are initially agreed and signed off the actual development work should be already started even a whole group of details are missing. In case the agreed scope / scope items will change so that it will cause re-work or the whole big development effort the changes need to follow change request process and cost estimates to be updated.

M2 and M3 milestones can be a check points in the project or Release process but those milestones do not require additional arrangements and development works can be continued until the official acceptance testing is done in QA environment. Software under acceptance testing is official Software proposal what is delivered for the testing environments (TEST and QA) at the same time.

Test Data for the testing is copied from the production environment and scrambled so that none of the details can be identified or used in a harmful way. Data copying cycle could be following the release cycle and be done every time when new release is about to start. Data copying process should be finalized before any testing can be started. Test Data can be copied for all testing environments and copying process and data scrambling script can be automated.

Quality Assurance and software testing activities will continue until final acceptance is done (M4 milestone).

The product is ready for production deployment activities. Before the official production usage starts the smoke test of the delivered product will be executed successfully in production environment.

Before closing the project or release in M5 milestone lessons learned will be collected in post retrospective / lessons learned session and analysed. Analyzed results will be used to improve the way of working in a next release.

Project documentation should be re-organized so that all the important project related documents will be found in one place (latest versions only) under the project page by release.

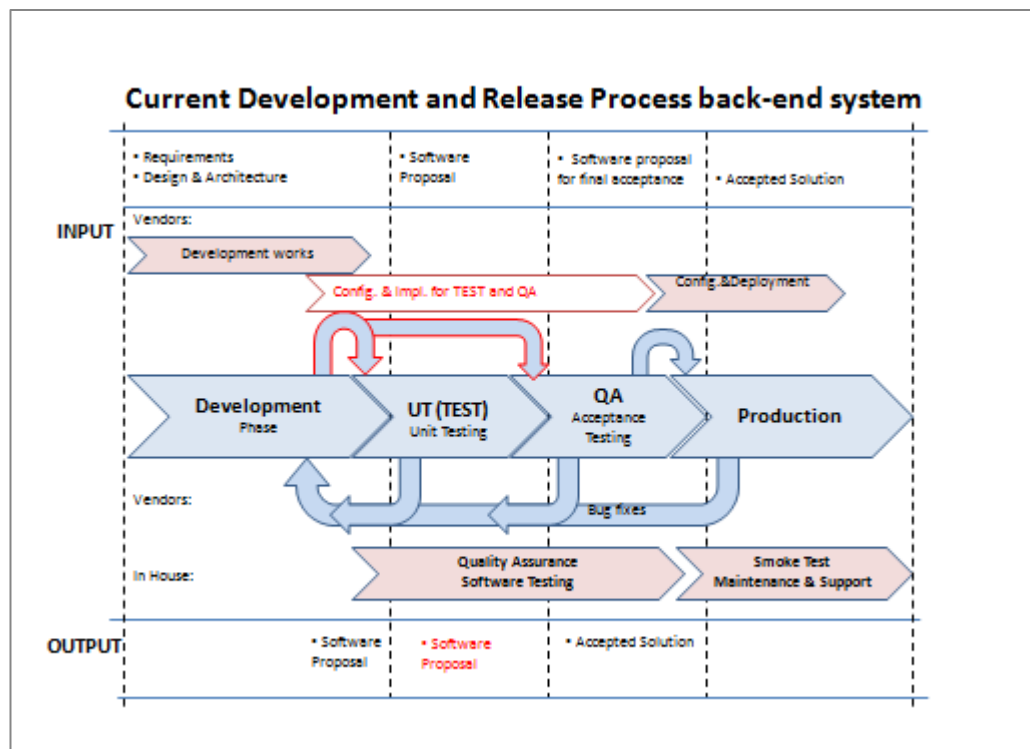


Figure 16. Proposed Development and Release Process

Release management is the process of managing software releases from development stage to software release in production. Currently there are three different phases before software release is ready to move to production environment for the real production usage of end-user.

The prototype what was piloted and what is the proposed solution for Release-process remains in high level as a same as it was in prototype-phase. Purpose is to keep still the two testing environment before releasing the product to the production. By this some flexibility is got for performing in the future performance and security testing. These two environments also give some flexibility especially when the fastening release process or in case it already works very fast.

There are many improvement suggestions what can be piloted and then decided if those would be valuable to change them in the process.

When the performance and security testing are integrated as a part of the testing process and practices the usage of environments can be slightly adjusted. The time when regression testing automation is in place (for ERP and WEB and CRM) these two separate environments give some flexibility in scheduling. Useful would also be if more automation scripts could be taken into use to help on configuration work and in software deployments for different environments and branches.

With a current practise where the vendor do software verification in an own testing environment there are at minimum two ways to approach in the future to shorten the deployment cycle; either software proposal is delivered for the both testing environments (TEST and QA) at the same time or the vendor will move the development work for the current TEST server and client side testing team can start the initial testing earlier. When software proposal is verified in TEST by the vendor and initially tested by the client it can be deployed in QA environment for the final acceptance.

The suggestion now is to do the deployment for the both environments at the same time and in H2 2014, start discussing the option two with the vendor.

One suggestion / option is to develop further the release process is to separate different processes and practises for new development work in Releases and for bug fixes.

Deeper co-operation with the vendor/-s would give the common advantages for all for example by sharing the automation test cases with vendor/-s.

In test management planning board different type of issues can use separate processes. Bug fixes without deep dependencies for any other modules can follow the same approach as document deployment, planning in kanban board and new development works in one release will follow the same procedure as currently, planning is done in agile planning board. Automation testing can be taken into use in regression testing and the scripts can be placed to the script library.

Manual Testing i seen as an opportunity to react very fast for the missing functionalities especially when more are talked about exploratory type of testing where the purpose is to find more exceptional type of test scenarios. There will be lots of room for manual testing but the composition of testing will change.

By centralizing testing activities the overall Lead Times can be reduced because the work would become repeatable except release development and bug fixes. The proposal is to shorten the release cycle time from 1,5 months to 3 or 4 weeks, Without proper automation this is much more challenging exercise than having the automation in place but on the other hand all the unnecessary steps will be disappearing when no time to repeat those.,

Business requirement gathering process should be made as a systematic process; done xx times a year. Business requirements should be collected from the business decision makers who give the directions in high level, end-users who give the requirements for the usability and functionality or it can be partner channels or authorities from where the requirements are coming. It would be ideal to place the requirement gathering in high, strategy level as a part of strategy planning process done twice a year. Business requirements which come from the business strategies and from the higher level plans have a direct impact for the system landscape and architecture, performance and capacity capabilities.

There should be categories and priorities defined how requirements are prioritized. Small requirements from users can be added as an additional requirement to the scope.

Release documentation should be re-organized by releases in documentation storage. Re-organizing will give the clear structure for the documents and the traceability and visibility what is delivered in which release becomes better.

Test Strategy and Test Plan documents are not needed to create for every release because in many cases include quite statistical information (unless for example new techniques, tools or methods are implemented) but in the beginning of for example the project is recommended to define and create the approach unless the concept is already created and much same for all the projects.

6.2 Project Metrics and Measures to follow up

KPIs (Key Performance Indicators) – List of project measures - Proposals what kind of measurement (KPIs)

Many times in projects the designed product is planned into a domestic market but the working environment is international, global. This can mean that some of the suppliers are working in different countries and some parts are done in national area with domestic suppliers. For example web-part what is visible for the customers is done locally or at minimum translations are done locally for target country and market, integrations for the different systems can be done locally or be partly done in other countries and back-end software development is done abroad, mostly in Asia, in India.

The case project web-part and interfaces for different target systems were locally developed by different suppliers and core / backend system (ERP – Enterprise Resource Planning system) was developed in India. Quite often this can mean different way of working with different suppliers therefore it is important that project management level common processes, practices and metrics are defined clearly in the beginning of the project. Those to be agreed with different parties (supplies) and all the sides are following same practices and reporting requirements in the project. High level agreements can agreed in contract level and common practices are agreed within the project team.

Often the project level measures are based on by the project and by the customer. If the customer is Start-up Company and all the things are developed from the scratch these things to be created in the project planning phase. Different supplier practices to be aligned in a same phase and can be break down for project and team level.

Cost, Quality and Productivity including Time form together the foundation for the Project measures and KPIs what needs to be created.

In the presentation Kendrick (2009) presents Measurement Levels Strategic Goals, Strategy Map, Balanced Scorecard and/or Strategic Indicators, Program Management, Portfolios and Programs, Projects and Operation and Maintenance.

Map the measures by defining the measurement level and define the focus for it, for example in strategic and project, portfolio level and for life cycle phases. Here is an example: Measurement level is Project and Focus is How well projects are performing.

Projects have different areas and phases in which different metrics are defined. Those are: Project Planning and Control, Project definition, Issue Management, Risk Management, Scope /Change Management, Reporting and communication, Scheduling and administration, Quality Management and Financial Management.

In the presentation is expressed Measurement & Metrics in different project areas:

- Monitoring Project Time (Time Management Performance, Actual vs. Planned Duration, Total Time to Complete (by phase by task))
- Comparing Costs (Actual vs. Planned Costs, Original Estimates, Baseline changes)
- Measuring Product / Service Quality (Availability, Customer Satisfaction, Error Rates)
- Measuring Risk Management (Risk Indicators, Incidents, Impact on Budget, Schedule or Quality) and
- Corrective Action Plans (Project Initiative to Fix, Effectiveness of Corrective Actions).

**Using measurement & metrics:
Earned Value Management**

001

EVM integrates:

- Scope
- Schedule
- Cost
- Resources
- Technical milestones

Measures project performance

Compares planned to actual costs and schedule

Estimates final project costs

Measures & Metrics (c) 2009 kendrick@p2c2group.com 26

Figure 17. Project Management Office Summit: Measures and Metrics for PMO success, (Kendrick, 2009).

Hobbs (2007) presents the list of the PMO functions There is the comprehensive list about the things to be followed periodically in the projects, below is the list of five selected, important functions:

- Report project status to upper management
- Monitor and control project performance

- Develop and maintain a project scoreboard
- Conduct post-project reviews
- Implement and manage database of lessons learned

Project metrics should be defined and be the same in the different projects, measuring should be done systematically, followed periodically and improved continuously for the next project. Project Results can be compared at program and portfolio level.

Visualizing the project metrics and measures helps on communication with the project team, stakeholders and business owners. All the measures should be analyzed and metrics need comparison. Projects need baseline what will only be adjusted when project scope is changed. Metrics should include target level and industry benchmarks, historical performance should be shown as a trend metrics.

In the presentation there are two examples of metrics dash-boards one is for visualizing performance metrics in the project and the other example is about visualizing balanced scorecard metrics. Project metrics and Balanced scorecard metrics can partly the same or cumulative figures from all the projects. Following example is balanced scorecard metrics dashboards. Colours are indicating how individual metrics is performed against the set target.

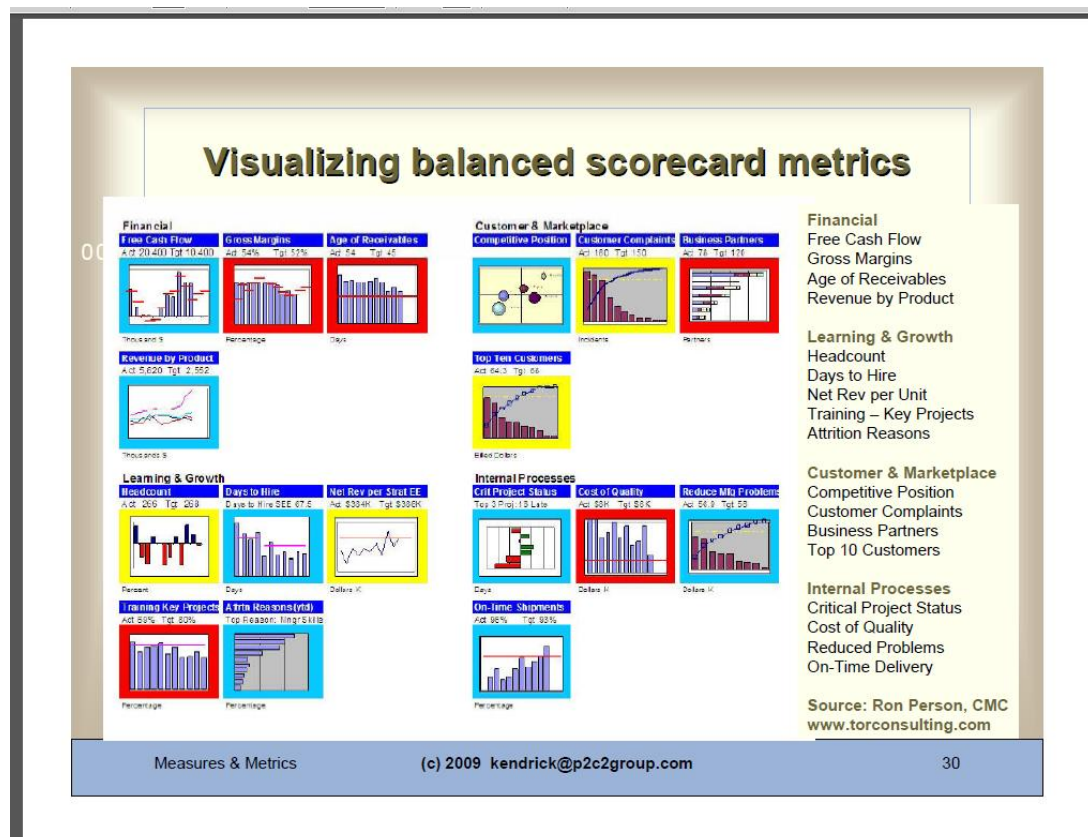


Figure 18. Balanced scorecard metrics / Hobbs (2007)

Another example for Quality Assurance and Project Management is presented in IT Knowledge Exchange forum, suggesting project measures Project Budget, Project Composition, Project Estimation, Project Management, Project Resource, Project Team, Software Development, Software Project, Software Release and Team Composition from cost and time wise.

In the Appendix 1 is collected strategic targets in different project areas (planning, requirements and priorities, project management, scope, schedule, costs risk management, software quality, resources, change management, software development, communication, meeting practices, documentation, stakeholder satisfaction, go-live activities, project tools, technical choices and implementation and architecture. These questions can be used to help for example internal audits, or following and clarifying the project status and what has been done already, when and what things are still missing. In the column key performance indicator is put the date when the action is checked.

Ten biggest mistakes in managing organisational performance, (Barr, 2014). Six mistakes are chose as most important ones:

Mistake #1: Rely just on Financial statements

Often in the projects are measured financial figures like profit and loss, revenue and expenses but these measures as such are too little and too late. It is also important to measure customer related information like customer satisfaction, customer loyalty and customer advocacy. Better approach would be to include customer related measures for the metrics and to see in advance what coming and when there are unacceptable results is coming. By this there is chance to influence the situation before it has actually happened already.

Mistake #2: Look only at this month, last month, year to date

Most of the financial performance reports summarise the figures by values. There are at least four different types of values: 1. Actual by current month, 2. Actual by last month, 3. % variance between them and 4. year to date. Would it be better to have reliable reports telling when really the problem was there and needed the attention instead off wasting time and effort chasing every single variation?

Mistake #3: Setting the goals without ways to measure and monitor them

In most organizations business planning process is well established through the different management levels of the organization interesting finding is that these goals and objectives are not measured well. Following example shows concretely how "measure" is established:

"Implement a customer relationship management system into the organisation by June 2006 (for a goal of improving customer loyalty)"

This is not a measure - it is an activity. When the goal is measured well it would be an evidence of how much customer loyalty the organization had and the organization is not wasting the valuable effort and money without real feedback.

Mistake #4: Using tables instead of graphs to report the performance

Common practise in many companies is to show the performance in tables instead of graphs. Tables concentrate focusing for the point of data instead of showing the pattern and visualizing the pattern in a trend graph.

Mistake #5: Failing in identifying how performance measures relate to one other

Decision makers should be aware of how different measures relate to each other's otherwise it is quite impossible to find the real root causes for poor performance or to improve the performance in a short period of time.

Mistake #6: Collecting too much useless data and not enough relevant data

Data collection requires time, effort and money and it generates costs. The best investment for data collection is got when collecting the correct data and enough (not too much useless data and not enough relevant data) for the decision makers. The decision makers cannot do the right decision with insufficient or irrelevant data, only wrong decisions or assumptions can be done.

6.3 Improvement based on the test / Lessons Learn

6.3.1 Corrective action list / Project Management Practises

People who are involved for the current daily release practices should continue the continuous improvement work of "WaterScrumFall" process.

Project Portfolio Management Tools are defined. JIRA is having a main role as a project portfolio management system. Different projects have been created for the system for different areas. There are projects for the back-end system (ERP), web-solution, partner channels etc. The tool gives the visibility to follow different type of issues in a structured way in the system. New features and Improvements are following the same practises. Test Management is integrated as a part of the same tool. Further development of the current practises will continue.

In the continues project model and releases the scheduling and prioritizing practises are in place. Requirements from the different sources are prioritized based on the business needs by a key group of people. There are clear priorities for critical and major development works. Proposed time schedules for the half year or the year are agreed with the vendor. Agreed high level time and priority schedule is communicated for the different stakeholders.

Further development work of reporting, measuring and visualizing the measures will continue. Management in different levels should have relevant information and correct, fact based data available for the decision making. Reporting practises should be systematically done, correct reports should be available and also report creation practises should be organized and controlled so that the end won't be hundreds of useless reports.

Tool and process for Issue and Error management is selected and in daily use. Also Change Management and Improvement (New Features) practises and roles are in place. Just recently Test management tool is chosen, piloted, metrics defined and one release delivered by using the developed procedure. Further development will be done together with the suppliers and agree roles and responsibilities, additional for that different testing areas to be integrated as a part of testing practises.

Identified a lot of possibility to develop security testing practises. Security testing to be integrated for a project model and to be part of the project concept and release testing practices.

Even though Training practises are already piloted and integrated as a part of the delivery model further development. Scheduling of the activity to be part of the planning practises, the best option would be to integrate training activity as a part of testing ac-

tivities. Training planning is done when release planning is done, the user instructions are done together with the tester(s) and business responsible. By this arrangement the best coverage is got, IT changes and business usage are covered, official templates are used and creating several duplicate instruction creation is avoided.

When everything what comes to the processes, reporting, measurement practises and metrics is been made from the scratch there is a huge amount of opportunities to continuously improve things

Baselines should be implemented for the project and release planning practises. Deliverables will be measured against the defined baseline.

Projects and as a part of the release practises should start measuring Lead Times. Total Lead Time is the time what is consumed for the whole delivery process and part Lead Times together make the Total Lead Time. Part Lead Times can be measured one by one or in groups.

Development and release plans, schedules and deliveries from different vendors should be aligned together then the customer gets the product or part of the functionalities what can be working together.

IT Cost Management and cost follow up is only in high level and periodically reported to the top management and people working on in IT department it is not visible in high or detail level. Clear templates and instructions how top management would like to see payback figures in investment calculations are not share with employees in IT department.

Measurement practises and metrics are in-sufficient in project management.

Systematic collecting of lessons learned to be implemented. The collection activity can be non-bureaucratic activity. Lessons learned can be for example collected in post retrospective meeting when go-live for the project or release is done. Deming cycle to be utilized for the way of working (Plan -> Do -> Check -> Act). Corrective actions and continues improvement improve the both practises; processes and project management framework. People, both the project people and stakeholders will see all the time

how things are improved and how they are involved for the improving the processes and practises.

Deeper level cost management practises to be implemented, for example the cost estimations to be got by task / development item from the suppliers. This practise makes for the project manager /product owner possible to follow carefully the project / release costs and report or give the indications for the management in case some delays in the schedule or if the planned costs are exceeded.

Full Traceability and matrix is not in use from requirements to accepted solution, clear visibility from each of the requirement through the release process is missing. There are tools and practises implemented but repeatable practises are not really repeatable and not visible for the whole team. Couple of approaches can be re-tested, from requirements (business requirement) to release and from the user story (web development) to release.

Should agree the practises for non-functional testing as a part of testing practises. Performance and security testing practises to be integrated as a part of testing practises, testing scope and test cases to be defined and reporting to be done in test management tool.

6.3.2 Improvement ideas for Release Process - Back-end system

One option to develop further the release process is to separate different processes and practises for new development work in Releases and for bug fixes. Deeper cooperation with the vendor/-s would give the common advantages for all by sharing the automation test cases with vendor/-s. In test management planning board different type of issues can use separate processes for example new release planning uses agile planning board and independent bug fixes uses kanban methodology.

Business requirement gathering process should be systematic process done xx times per year. Business requirements should be collected from the business decision makers who give the directions in high level, end-users who give the requirements for the usability and functionality or it can be partner channels or authorities from where the requirements are coming. It would be ideal to place the requirement gathering in high, strategy level as a part of strategy planning process done twice a year. Business re-

quirements which come from the business strategies and from the higher level plans have a direct impact for the system landscape and architecture, performance and capacity capabilities.

There should be categories and priorities defined how requirements are prioritized.

Systematic handling process for bug fixes to be agreed.

Clear versioning should be agreed and version control practices implemented, release related naming conventions what are concerning documentation to be agreed.

When any type of the issue is identified in production environment it is analyzed what type of issue is and how urgent fix is needed. The systematic review process and practices for Production issues should be agreed also for lower level priorities and agreed process should be implemented. Each found issue is analyzed and prioritized, this practice gives the instantly the needed priority for blocker and critical issues and lower priority for not so high severity level defects.

Should develop the security testing practises and integrate it part of the release process.

In release process Total and part Lead Times are not followed systematically to improve the process. Systematically followed dates and times what have been spent for testing in each environment gives the visibility where we are now and what is the lead time in every and each step. When bottlenecks are identified those can be adjusted one by one. Total and part Lead Times will be reduced.

Roles and Responsibilities should be agreed in detailed level in testing.

Test automation tool what is compatible with different systems (i.e. ERP, CRM) should be taken into use. Test case sharing could be done between client and vendor. Full E2E (end-to-end) testing can be done when tool is compatible for the main systems.

Speed up the process development actions. Should reserve for example all together one day in each week to do some process development works (rather can be shared for different days, couple of hours timeslots for that).

By taking a critical watch for all the tasks what are part of the process some time savings can be found. Systematic way of working in many areas saves time when things are organized systematically.

IT Cost Management and cost follow up should be visualized also for people who are working on in IT department, not only for the top management.

Full Traceability and matrix should be utilized into use from the requirements to the accepted solution; clear visibility from each of the requirement through the release process is required. Tools and practises are implemented but repeatable practises are not visualized for the whole team.

Sprint/Release planning and management should be taken into use in all test planning, ERP and CRM.

Full end-to-end process has dependencies on external systems also in testing environments. When external services are not stable it causes delays for the testing. Last step of end-to-end test with banks and accounting firm should be integrated for the testing to get visibility what is tested and how in each release process.

Practises for non-functional testing as a part of end-to-end testing should be agreed. Performance and security testing practises, test scope and test cases in back-end release process should be defined.

7 Conclusions

7.1 Summary of the Project

The scope of Software Delivery Concept was covering an IT-project which IT-architecture consists of the following systems and integrations:

- Core System - **Back End system (ERP)** including different modules for business use including Service Center (Quotations, Sales, Accounting and Claims)
- **Partner channel system** what is available for different business partners, at first place for car dealers
- **WEB-front end** for consumers (individual customers)
- **Integrations** between these three systems (Back End, Partner Channel and Front End (WEB)) and

- **Integrations for external systems** (for example banks, authorities, bookkeeping system, credit check).

In the project there were three different suppliers and three different type of way to work. Some were following mostly or partly agile methodologies, some more traditional software delivery project approach.

Current process or some parts of it and current practices were aligned because the process was not providing sufficient visibility and information for the company management and for the project team.

GOAL and Target was to create a Software Delivery Concept for the back-end system (ERP), give a comparable metrics for the project management and make an improvement suggestion list of the future actions.

SWOT analysis was done for the Project Management practices and for the back-end Release process. Software Delivery Concept was created based on the SWOT analysis, Theme interviews and information gathered from different sources and from literature.

Software Delivery Concept proposal was piloted. Piloting and testing was done for the prototype solution. Got experience and outcome was analyzed. Outcome from the pilot was adjusted for improved Software Delivery Concept, example of Project Metrics and Measure in Project Management was presented to give an example how internal project follow up can be done. Improvement ideas and lessons learned from the project were collected into the corrective action list / improvement idea list was provided.

Out scoped were infrastructure, technical architecture and environment management, detailed level technical information and documentation, security and excluded external systems and details of cost management.

Key findings:

Proper communications for all stakeholders, selected tools, used working practices, metrics and continues improvement have an essential role to manage successfully Project management and Release processes and to deliver products successfully to the customers and end users.

Fact based and right timing communication is the most important thing in all projects and smaller releases. All stakeholders should get the needed information about the delivery scope, schedule and business value of the product.

Integrated tools and automation provide the needed efficiency for the approach. Aligned working practices shorten the cycle times and make possible to deliver most valuable product scope for the customers and best possible time alignment for the product delivery. Sufficient measures and metrics provide the needed visibility for the management, other stakeholders and for the project team to follow the activities and delivery progress. Continues improvement makes possible to learn from done mistakes and improve the deliveries every time.

There are some areas what can be further improved and measurement practices can be sharpening. Overall communication in the projects and releases are in good level but communication for the business side and for the end users can be further improved. Time schedules, delivery scope and needed actions from business side can be done more systematically, being a part of the delivery. Project timelines and cycle times can be measured in more accurate level to make the progress more visible for all and to improve the current processes. Full traceability matrix from the requirements to the delivery would make more visibility for all stakeholders. More automation should be implemented for the processes it selves and for the testing to speed up the release cycles. Project lessons learned practises would make the continues improvement more systematic and project / release closures closes the done deliveries.

Some highlights from the researchers:

Risa Fogel, Senior Managing Director at Cushman & Wakefield, advices to have three different topics to make the understanding better and communication easier.

Instead of talking technology and using technical terms conversations should be tied to the business goals. Establishing business-facing roles, business relationship managers, will help on communication to different stakeholders. The role is essential to explain business and technology challenges with equal clarity. Establishing the governance processes is the third topic. It will provide the visibility into decision making and spending.

Eric Winqvist, CEO from Jama Software gives very valuable advices and brings forth many things what are important and in which need to be carefully considered. Some of the examples are:

- Requirements are the number one reason for the delays
- Aligning the delivery processes and finding the core business value is essential - Thinking of how to do that?
- Start reusing and review early and often with the stakeholders to improve the speed and quality. The key is to keep the process iterative and small.
- Cycle times can be reduced by reusing the information from libraries and launch by launch updating the new information and improving the cycle
- The traceability through the project or delivery is essential

Forrester Consulting who has done the examination about organizations delivering the products express that product delivery garners lots of attention, it has been highlighted into a strategic role in many organizations. Forrester Consulting stated also that the priorities should be agreed clearly by stakeholders.

7.2 Evaluations of outcome against the target/object

The goal and the target was to create a Software Delivery Concept for the back-end system (ERP), give a comparable metrics for the project management and make a improvement suggestion list of the future actions.

Input and activities what are done for creating the Software Delivery Concept are SWOT analysis (Project Management practices and back-end Release process), Theme interviews and information gathered from different sources and from literature.

Current processes are the outcomes. They live and improve themselves based on the feedback from each of the releases. When something can be improved or done better those ideas will be taken into use in the next delivery.

Outcome from the pilot was adjusted for improved Software Delivery Concept, example of Project Metrics and Measure in Project Management was presented to give an example how internal project follow up can be done. Improvement ideas and lessons learned from the project were collected into the corrective action list / improvement idea list was provided.

Goal and targets for this project are achieved because current and improved processes and practices are described, bottlenecks are identified.

In a next release some of the improvement ideas will taken into use to achieve more efficient processes for the software delivery. When more accurate metrics will be taken into use more accurate results and impacts can be visualized and implications are more measurable.

References

- Ylä-Anttila, Pekka. (2012). *Finland's Transition to a knowledge Economy*. 13th August 2012.
Available: <http://blogit.jamk.fi/cfc/2012/08/13/finlands-transition-to-a-knowledge-economy/#.U4OxAXHyWM8>. Last accessed 15th May 2014.
- United Nations Radio. (2012). UNCTAD: *Finland leads the world in ICT employment*, 15th February 2012. Available:
<http://www.unmultimedia.org/radio/english/2012/02/finland-leads-the-world-in-ict-employment/>. Last accessed 19th Nov 2012.
- Finland's Information Society, (2003). *Public-Private-Partnership Works Well in Finland - Article of the month*. Available:
<http://e.finland.fi/netcomm/news/showarticle2dc0.html?intNWSAID=15229>. Last accessed 19th Nov 2012.
- Greengard, Samuel. (2014). *Six Top Tech Trends to watch in 2014*. Available from CIOInsight: <http://www.cioinsight.com/it-news-trends/six-top-tech-trends-to-watch-in-2014.html/>. Last accessed 15th May 2014.
- Fogel, Risa. (2011). *Tree Steps to Improving Business - IT Alignment*. Available:
<http://www.cioinsight.com/it-management/expert-voices/three-steps-to-improving-business-it-alignment.html>. Last accessed 19th April 2014.
- Winqvist, Eric . (2011). *Trends Shaping Product Delivery*. Available:
<http://www.jamasoftware.com/>. Last accessed 15th March 2014.
- Winqvist, Eric . (2014). *Modern Product Delivery*. Available:
<http://vimeo.com/95053850>. Last accessed 20th May 2014.
- Flinders, Karl. (2011). Failed IT projects demolish big businesses and executive careers, says researches. Available:
<http://www.computerweekly.com/news/2240105403/Failed-IT-projects-demolish-big-businesses-and-executive-careers-say-researchers>. Last accessed 15th May 2014.
- Centreline Solutions Inc. (2005). *Ten Major Causes of Project Failure (02.18)*. Available:
http://www.google.fi/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=1&ved=0CCsQFjAA&url=http%3A%2F%2Fwww.gtislig.org%2FDocuments%2F10_Major_Causes_of_Project_Failure.pdf&ei=F8KDU_-TLtTc4QTCp4DIBg&usg=AFQjCNFQy6haD9OtHrJ47X-BttDupPC7Qg. Last accessed 15th May 2014.

- Zhu, Pearl, (2014): Five ponderings on Why IT Projects Fail. Available: <http://futureofcio.blogspot.fi/2012/01/five-pondering-why-it-projects-fail.html>. Last accessed 19th May 2014.
- Zhu, Pearl, (2012). *IT failure: Symptoms and reasons*. Available: <http://www.enterprisecioforum.com/en/blogs/pearl/it-project-failure-symptoms-and-reasons>. Last accessed 19th Nov 2012.
- IT Business Edge. (2014). *Why IT projects FAIL*. Available: <http://www.itbusinessedge.com/slideshows/show.aspx?c=81818>. Last accessed 19th April 2014.
- GTNews publication Afponline.org. (2014). *IT Project Success Factors*. Available: <http://gtnews.afponline.org/articles/7272/1.gif>. Last accessed 19th May 2014.
- Winkler, Colleen. (2010). *IT Project Success Factors* Available: <http://colleenwinkler.com/2010/09/22/lets-get-smart> . Last accessed 19th May 2014.
- Bauer, Martin, (2008). How to plan CMS project. Available: <http://www.martinbauer.com/Articles/How-to-Plan-a-CMS-Project/Project-Success-Factors>. Last accessed 19th May 2014.
- Dr. Brian Hobbs, *The Multi-Project PMO: A Global Analysis of the Current State of Practice*, whitepaper prepared for Project Management Institute, 2007; page 22.
- Wieggers, Karl. (2014). *Project management Best Practises: Foundation and Planning*. Available: <http://www.jamasoftware.com/blog/project-management-best-practices-foundation-planning/>. Last accessed 15th May 2014.
- Kendrick, Jim. (2009). *Project Management Office Summit: Measures and Metrics for PMO success*, P2C2 Group Inc: page 26.
- RallySoftware, (2014). *PMBOK Practices mapped to Agile Practices, Integration Management A Collaborative, Iterative Process*. Available: <http://www.rallydev.com/>. Last accessed 9th May
- RallySoftware, (2013). *The Project Manager's Agile Survival Guide*. Available: <http://www.rallydev.com/>. Last accessed 9th May 2014.
- Agile Alliance and Institute Agile, (2013). *"Subway map"*. Available: <http://www.agilealliance.com>. Last accessed 9th May 2014.
- Anderson, David (2010). *Kanban, Successful Evolutionary Change for Technology Organisations*. Available: <http://www.google.fi/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=2&ved=0CDUQFjAB&url=http%3A%2F%2Fwww.scrumsense.com%2Fwp->

content%2Fuploads%2F2010%2F02%2FKanbanManuscript-dja-rev1_07_2.pdf&ei=9KyEU5vKCor34QTO4IHADw&usg=AFQjCNHGRG4zHGcQRQgtTI3XNYyMHE gdRw&bvm=bv.67720277,d.bGE. / Last accessed 9th May 2014.

ASQ Quality Tools, 2014: *Learn about Quality: Benchmarking, Balanced Scorecard, Plan-Do-Check-Act*.

Available: <http://asq.org/learn-about-quality/quality-tools.html>. Last accessed 9th May 2014.

Simpson, Dorian. (2011). *Five Challenges to Agile Planning*. Available: <http://www.jamasoftware.com/>. Last accessed 15th March 2014.

StarkInfoTech. (2014). *Software Testing Life Cycle (STLC)*. Available: <http://www.starkinfotech.com/software-testing-life-cycle-stlc/>. Last accessed 9th May 2014.

Software Testing Software, (2014). *Software Testing Software*. Available: <http://www.softwaretestingsoftware.com/>. / Last accessed 9th May 2014.

Webopedia, (2014). *Regression Testing*. Available: http://www.webopedia.com/TERM/R/regression_testing.html. / Last accessed 9th May 2014.

Microsoft Developer Network, (2014): *Regression Testing*. Available: [http://msdn.microsoft.com/en-us/library/aa292167\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/aa292167(v=vs.71).aspx). / Last accessed 9th May 2014.

Hudson, Tom, Smartbear, (2014). *Regression Testing*. Available: <http://smartbear.com/products/qa-tools/what-is-regression-testing/> Last accessed 9th May 2014.

Syncplicity. (2014). *Securing Data in cloud. pages 2-11*. Available: <http://www.itbusinessedge.com/slideshows/show.aspx?c=96494>. Last accessed 9th May 2014.

SoftwareDelivery, (2014). *Software Delivery*. Available: <http://www.softwaredelivery.com/>. / Last accessed 9th May 2014.

StackExchange, (2014). *What are the roles of Software Deliver Manager*. Available: <http://programmers.stackexchange.com/questions/138641/what-are-the-roles-of-a-software-delivery-manager>. / Last accessed 9th May 2014.

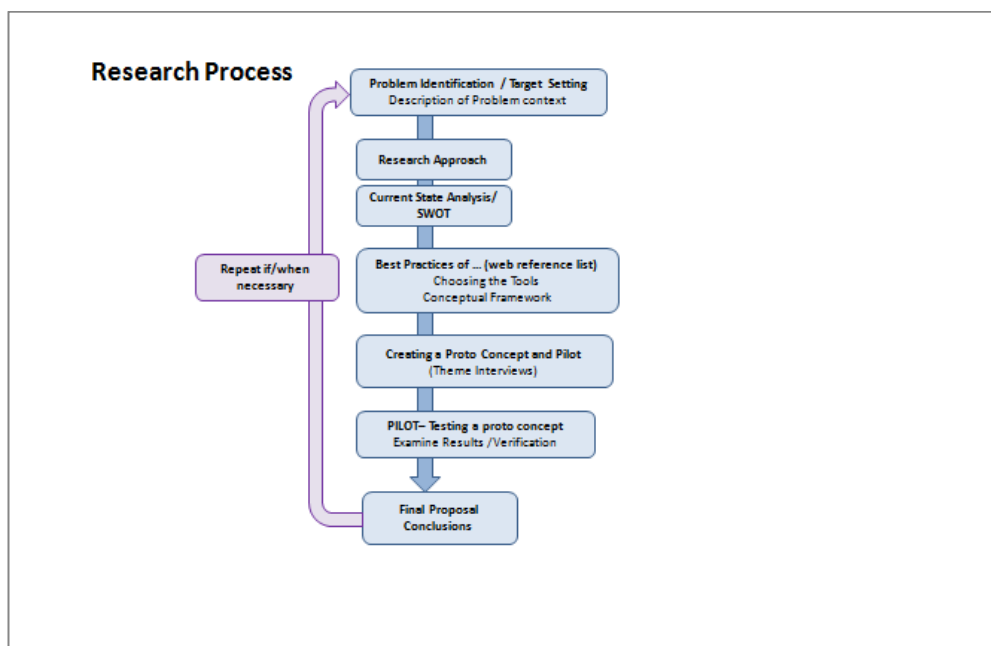
Jama Software (2011). *Five challenges to Agile Planning*. Available: <http://www.jamasoftware.com/>. Last accessed 15th March 2014.

IBM, (2014). *IBM Rational solutions for software and system delivery*. Available: <http://www-01.ibm.com/software/rational/strategy/>. / Last accessed 9th May 2014.

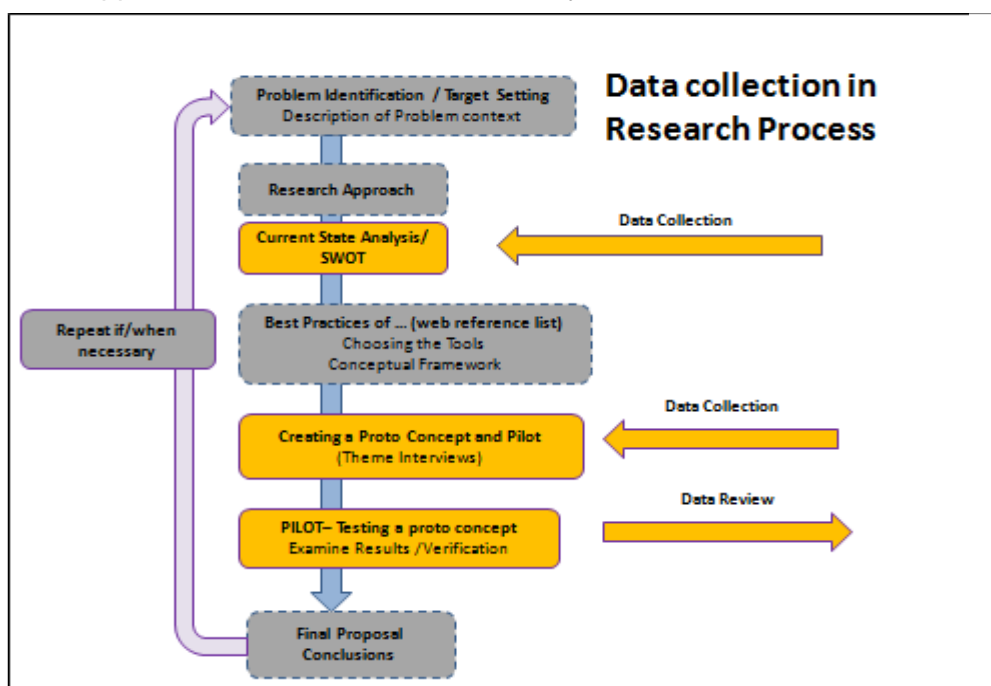
- Shah, Sunil, (2008). *IBM - Global Delivery Excellence*. Available: http://www-07.ibm.com/in/events/swu08/pdf/day1/itservices/global_delivery_excellence.pdf. / Last accessed 9th May 2014.
- Janzen, Nancy (2013). *StaffReady - What software Delivery Model should I choose?*. Available: <http://srdev.maplewoodsoftware.com/tag/software-delivery-model/>. / Last accessed 19th May 2014.
- Mordo, Aviran, (2013). *The Road to Continues Delivery - Part1*. Available: <http://www.aviransplace.com/2013/03/16/the-road-to-continues-delivery-part-1/>. / Last accessed 9th May 2014.
- Humble, Jez, Farley, David, 2011: *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*: pages 4, 25, 27, 38, 50, 86, and 187.
- Hobbs, Brian (2007). *The Multi-Project PMO: A Global Analysis of the Current State of Practice*: page 22.
- Khanduja, Jaideep, (2011). *Project Management and Project Estimation* in IT Knowledge Exchange forum. Available: <http://itknowledgeexchange.techtarget.com/quality-assurance/project-management-and-project-estimation/>. / Last accessed 19th Nov 2012.
- Competitive Solutions Inc., (2013): web-seminar "*Business Scorecard Best Practices for a Successful 2013*"
- 2014.Barr, Stacey. (2014). *Ten biggest mistakes in managing organisational performance* . Available: <http://www.staffready.com/2013/07/07/what-software-delivery-model-should-i-choose/>. Last accessed 9th May 2014.

Appendixes

Appendix 1: Research Process (2 Research Approach)

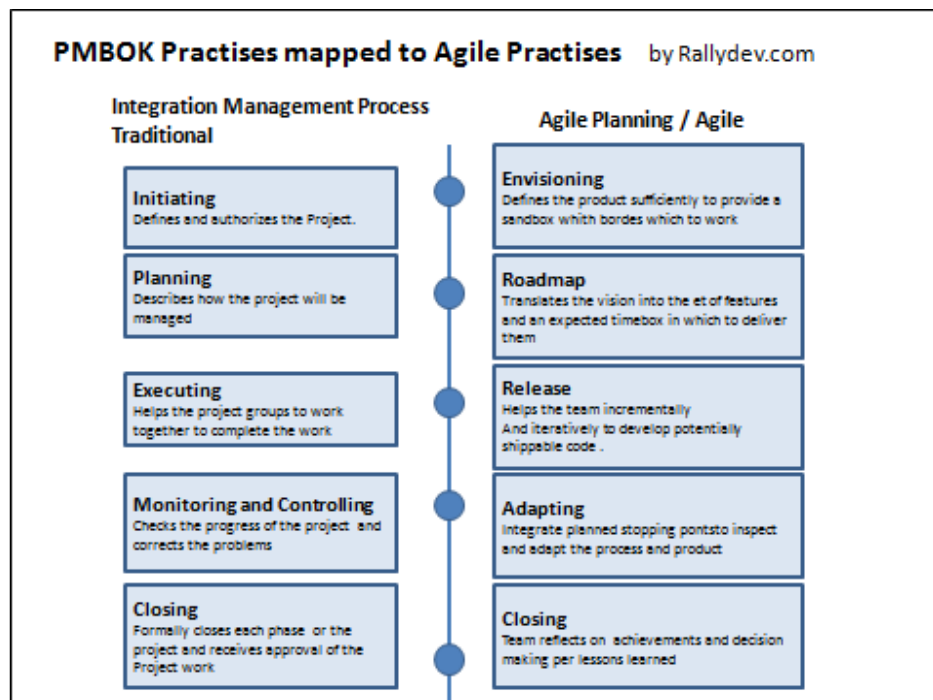


Appendix 2: Data Collection and Analysis

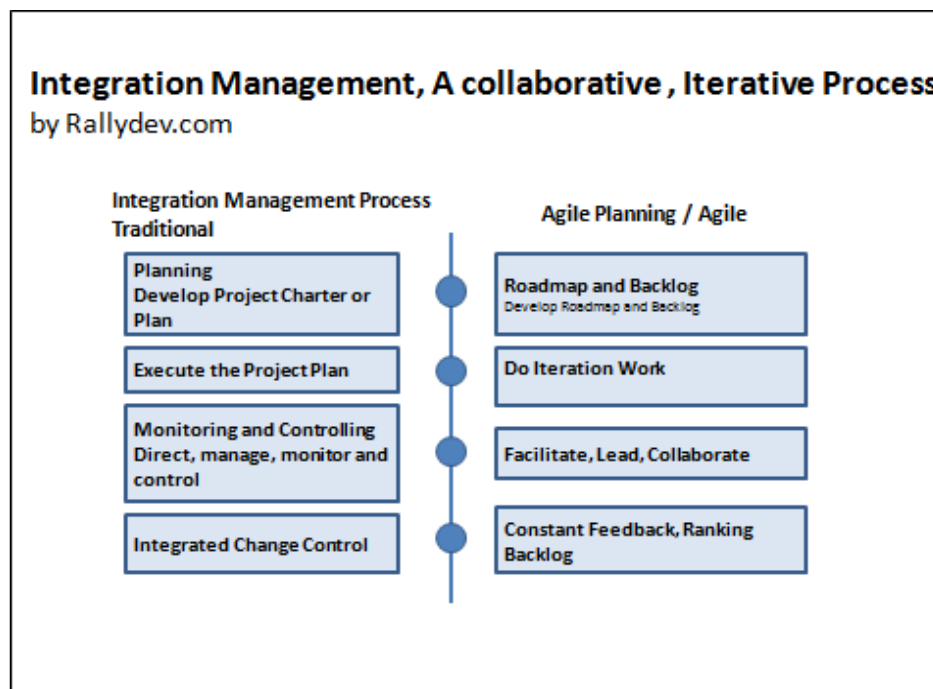


Data Collection in Research Process

Appendix 3: PMBOK Practices mapped to Agile Practices by Rallydev.com



Appendix 4: Mapping Integration Management to Iterative Process



Mapping Integration Management to Iterative Process in Traditional and in Agile Practices by Rallydev.com

Appendix 5: Two Software Delivery Models

	On-Premises	SaaS
Installation and System Maintenance		
Software location	On the network or desktop on-site	On Hosted Vendor servers in the cloud
IT Support	Required	Nor required
Additional Hardware	Yes	No
Implementation, Training and Support		
Consultative Analysis and set-up customization	Not Likely	Likely
Software upgrades	Extra Fee	Included
Employee Access	May not be accessible offsite	Accessible for administrators and employees
Scalability	Complex	Straightforward
Pricing	Upfront software and hardware purchase	Monthly/Yearly Subscription
Budget	Capital Expense	Operational Expense

Appendix 6: Project, Release and Testing questions

Project:

What are the key success factors (3 to 5 pcs) in your projects for on-time product / software delivery?

What are the things(3 to 5 pcs) causing delivery delays or confusion in the projects?

How long time is the release cycle in your process/project (from development to release in the production)?

How do you prioritize the requirements? Who do it and based on what priorities?

How do you analyze the dependencies between requirements and development items?

How costs are followed:

- a) at project portfolio level and periodically
- b) at project level and monthly bases?
- c) more detailed level and more often? Explain how.

Think of your last project. What are the things (3 to 5 pcs) could be done better?

Release process:

What are the five key things to speed up the release life cycle?

Is there a same release process for new development works, configuration changes and data-base changes?

Are all clubbed together or different approaches and different life cycles? If any of the previous processes differs from each others, explain how. How the planning is done?

When to use incremental approach or always? Any difference in front-end and back-end development?

Testing:

What tools are used for test management tasks, for issue management, for automation, for documentation?

How big part of testing is done with automated scripts in development and in further testing phases?

(for example: list the testing phases and percentage of automation) ?

Any examples of automated acceptance testing or partly automated acceptance testing?

What type of security testing is done in your last project? Who did it and who analyzed the coverage?

How much automation used?

What type of performance testing is done in your last project? Who did it and who analyzed the coverage? How much automation used?

How regression testing is done, manually or automated?

If testing is out-sourced how costs are followed?

- a) in a project level or
- b) separate cost management for testing activities?

Appendix 7 - 12. Theme Interviews

Theme Interview 1

The role of the interviewed person: SW Lead Developer Interview held: 9.2.2014

Topics: Project Management, Release Process and Testing

Your projects, last project, followed methodology: Agile/ Scrum or traditional Project Mgt

Project:

- Key success factors for on-time product / software delivery: continuous follow up - follow up or continues communication in the project team, efficient review process, design checking at every stage, continues integration, testing releases, learning from the mistakes.
- Things causing confusion and delivery delays in the projects: different type of changes especially user interface changes (development works and UI changes) - unclear or misunderstood requirements, too high level planning in sprint planning, too many changes, following scrum only in planning
- Release cycle is usually 6 months. - Business Requirements are prioritized by project manager based on what is the most important features. Information is gathered by benchmarking, from the user experience, from the market.
- Costs are followed at portfolio level and periodically. Costs are based on the resources / resource pools program.
- The things what could be done better / Lessons learned were better/prior planning, more detailed level planning - sufficient information for the development work reduce rework and re-cycles, requirement analysis, more detailed level sprint planning.

Release process:

- Five things to speeding the release process: continues integration builds, test automation, sprint reviews, nominated person checking the daily builds, performance testing is done before certification testing, review practises, Alpha Beta releases - final release to follow the same process.
- Same process is followed for all the changes, most of the features are in release, increments will have only the minor changes

Testing:

- Continues Integration tool, - Automated regression testing , automated unit testing, separate performance testing team do performance tests, separate manual security certification team do security testing manually
- Testing is in-house and done by separate testing team, costs are followed by project level.

Theme Interview 2

The role of the interviewed person: SW Program Manager Interview held: 14.2.2014

Topics: Project Management, Release Process and Testing

Your projects, last project, followed methodology: Agile/ Scrum or traditional Project Management

Project:

- Key success factors for on-time product / software delivery: stakeholder (business owners and users) involvement, good communication within the project team, project planning, understanding business requirements and regular follow up for the activities, efficient review process, collaborative team work.
- Things causing confusion and delivery delays in the projects are unclear requirements, scope changes, time differences, many vendors, set dead lines for the things and amount of scope items.
- Release cycle depends on the vendor.
- Business Requirements are prioritized by business owners based on the business need and most important features.
- Dependency analysis is done in big programs and projects in high level by project (planning efforts) and in technical level by vendors.
- Costs are followed in program and in project level monthly bases and reported to the needed stakeholders.
- The things what could be done better / Lessons learned were requirements planning what took all together more than 2 months, requirement prioritization.

Release process:

- For the release process not much visibility but speeding up the process clear business requirement analysis would help. - All the changes are coming in one release. Deployments are done either site by site or in some sites parallel depending on the project or programs.

Testing:

- The tools what were used for test management and issue management tasks: QC, for automation nothing, for documentation project workspaces, sharepoint.
- Manual regression testing was done and performance testing was done, but amount of concurrent users were in-sufficient, security testing was not done, automation was not used in testing but that was good thing to do.
- Testing is partly out-sourced to vendors and costs are followed project level, periodically.

Theme Interview 3

The role of the interviewed person: SW Project Manager Interview held: 14.2.2014

Topics: Project Management, Release Process and Testing

- Your projects, your last project, scope, followed methodology: Agile/ Scrum or traditional Project Management

Project:

- Key success factors for on-time product / software delivery are communication, clear scope/ clear requirements, project planning and scheduling, project follow up, roles and responsibilities, lessons learned.
- Things causing confusion and delivery delays in the projects are communication if it is not done properly, change management, unclear scope and business requirements.
- Release cycle is from four to six weeks.
- Business Requirements are prioritized by business owners and by steering group based on the importance, value, cost and resource savings.
- Costs are followed in projects real time, weekly and monthly level and reported to the needed stakeholders.
- The things what could be done better / Lessons learned were better communications for all stakeholders, change management.

Release process:

- There isn't any separate functions for the release process, work is done within the team but later, additional requirements are causing continues changes for the original plan. There has been discussions to freeze the scope and new requirement would be better to take for the next release.
- All the changes are coming in one release. Deployments are done either site by site or in some sites parallel depending on the project or programs.

Testing:

- The tools what were used for test management and issue management tasks: QC, for automation nothing, for documentation project workspaces, sharepoint.
- Testing is done by the team and by the business manually, no automated regression testing, no performance testing, no need for separate security testing.
- Further testing is done by users and out-sourced testing is not used.

Theme Interview 4

The role of the interviewed person: SW Product Owner Interview held: 11.3.2014
13.3.2014

Topics: SWOT analyses (Project Management and Release process), Project Management, Release Process and Testing

- Project - Improvements / Lessons learned

SWOT -analysis: Strengths, Weaknesses, Opportunities, Threads

Current State Analysis Project Framework, Current Project Framework-process

Current Release Process (back-end)

Project:

- Key success factors for on-time product / software delivery are good communication, clear requirements, project follow up, roles and responsibilities, detailed level scope item planning, prioritization.
- Things causing confusion and delivery delays in the projects are in-sufficient communication, sign off process with many iteration rounds, requirement changes and change management, unclear business requirements.
- Release cycle depend on the vendor and plans, can vary from 3 weeks to 1.5 months or 2 months depend on the plans.
- Business Requirements are prioritized by business owners and by product owner and vendor based on business need from the company management, market and from the users, based on the criticality.
- Costs are followed in projects level monthly bases and reported to the needed stakeholders.
- The things what could be done better / Lessons learned were better scope item prioritizations, better transparency, communication, training.

Release process:

- All the changes are coming in one release.
- web-development and interface development is following scrum, according to the agreed scope items and schedule

Testing:

- The tools what were used for test management and issue management JIRA, continues test automation tools, for documentation project workspaces, confluence
- Testing is done by the team and by the business manually, partly automated regression testing, performance testing periodically, initially planned security testing.

Theme Interview 5

The role of the interviewed person: SW Architect Interview held: 12.3.2014

Topics: SWOT analyses (Project Management and Release process), Project Management, Release Process and Testing, Project - Improvements / Lessons learned

SWOT -analysis: Strengths, Weaknesses, Opportunities, Threads

Current State Analysis Project Framework, Current Project Framework-process

Current Release Process (back-end)

Project:

- Key success factors for on-time product / software delivery are good communication, clear requirements, good quality, project follow up, .
- Things causing confusion and delivery delays in the projects are unclear scope and business requirements, many iterations before requirements are signed off, additional changes.
- Release cycle depend on the vendor and plans, can vary from 3 weeks to 1.5 months or 2 months depend on the plans.
- Business Requirements are prioritized by business owners and by product owner and vendor based on business need from the company management, market and from the users, based on the criticality.
- Costs are followed in projects level monthly bases and reported to the needed stakeholders.
- The things what could be done better / Lessons learned were better scope item prioritizations, better transparency, communication for business, training, retrospective after go live.

Release process:

- All the changes are coming in one release.
- web-development and interface development is following scrum, according to the agreed scope items and schedule

Testing:

- The tools what were used for test management and issue management JIRA, continues test automation tools, for documentation project workspaces, confluence
- Testing is done by the team and by the business manually, regression testing partly automated, unit testing automated, automated also some areas of system testing , performance testing periodically.

Lessons learned: better communication, reorganizing the documentation, speeding up the release process, exact information, fact based, reporting

Theme Interview 6

The role of the interviewed person: SW Project / Program Manager

Interview held: 18.3.2014

Topics: Project Management, Release Process and Testing

- Your projects, last project, followed methodology: Agile/ Scrum or traditional Project Management

Project:

- Key success factors for on-time product / software delivery: proper communication for all stakeholders. proper project planning and scheduling, best possible resource in the project, clear business requirements, project review and follow up and proper coordination of activities.

- Things causing confusion and delivery delays in the projects are: not finalized business requirements, changes in scope, unrealistic project timelines, lack of allocated resources and incomplete testing.

- Release cycle depends on the scope, project and vendor, from 4 to 6 weeks or in some cases can be longer.

- Business Requirements are prioritized by management and business owners based on the urgency, business need and based on the value for business.

- Costs are followed in program and in project level monthly bases and reported to the needed stakeholders, business owner and in big projects to the steering group.

- The things what could be done better / Lessons learned were requirements analysis, testing, prioritization, change management.

Release process:

- Most of the changes are coming in one release, in some cases some of the complex items can be moved for the later releases, different type of changes (configuration, development work) are combined to the one release or delivered separately depending on project scope, both automation and manual work are needed, multi environment model (development is done in own server)

- Shorter and faster process for the small fixes is not used

Testing:

- The tools what were used for test management and issue management tasks: QC, for automation QTP, for documentation project workspaces, sharepoint.

- Regression testing is mostly automated, performance testing is done in some cases depending on the project, scope and need, security testing is not done.

- Testing is partly out-sourced, acceptance testing is done by customer and development work related testing is done in vendor side.

Appendix 13: Questionnaire for project work

Strategic Target	Action (= the work to be done)	Questions to help clarify
Project Planning	Project Planning	Project Planning
	Define Business purpose of project - well-defined and communicated,	What kind of project plan exists?
	Define Business Plan in structured or detailed model	How the plan is validated?
	Define the technical solution, resources, skills	Is the project plan clearly structured?
	Do Planning (allocating time to design, testing and requirements gathering)	Are the project goals clearly defined?
Estimates	Estimates (Planned / Actuals)	Are the decision points defined?
	Do the base lines for the estimates in the different areas	Is there a baseline defined for the project?
	Divide the Project into the work units (teams)	Is the project divided into work units (teams)?
	Involve the different suppliers to the plans and into the big picture	How the different suppliers are involved and their plans are part of the big picture?
	Do also the project Planning for the following areas: Requirements - Scheduling - Scope and objectives	Is the following areas part of the project plan? - Requirements - Scheduling: - Scope and objectives?
Requirements and Priorities	Requirements and Priorities	Requirements and priorities
	Plan and do the Requirement gathering Make sure that the requirements are clear, complete and consistent	Are the requirements clear, complete and consistent?
	Make sure that requirements analysis process is defined. Do Requirements analysis.	How requirements have been gathered from the business?
	Be careful that Over Specification won't happen.	How requirements are analyzed? What is the requirements analysis process?
	Prioritize the requirements. Define the Business Critical Requirements.	Are the requirements prioritized? How the prioritization has been defined? Business criticals are priority 1. What are the prioritization criterias?
	Check integration with other corporate processes (e.g., procurement, strategic planning).	
Project Mgt	Well-managed Project Execution - expectations defined.	Managing the project and project practices
	Define the Project methodology to follow, measures and metrics - use the project measures	How do you measure the progress?
	Follow the project progress regularly and sufficient level	What are the other measures?
	Check that Prioritization of projects is done.	How do you monitor project?
	Check that Project portfolio techniques defined	
	Check that Continuous improvement practices defined,	
Roadmap		
Schedule - Time	Schedule - Time	What scheduling tools are used?
	Scheduling tools and templates defined	How scheduling is done?
	Timely Delivery - Planned / Actual	How the scheduling is done from project schedule to the different areas? i.e. development, testing
Cost	Cost	Cost
	Define Acceptable ROI	How do you follow project costs?
	Make sure that project is staying within the Budget	License costs
	Follow the cost calculations (planned / actuals) - license costs - server costs	Server costs

Quality	Quality	
	Make sure that Quality Requirements are defined. Good Quality: Project Management is following the 3C Rules (communication, coordination & collaboration),	How it is ensured that solution meet quality requirements?
	Follow 3 C practises and measure the results in the project	
Resources	Resources	Resources
	Allocate the resources	Resourcing done?
	Follow the allocated and actuals	Allocations? How do you follow?
	Allocate also in task level	How tasks are allocated?
	Define the Roles and Responsibilities for each of the project members	Are R&R clearly defined with in the project team?
	Define the Roles and Responsibilities for each of Suppliers	Are R&R clearly defined within the different suppliers?
	Check that Project Team has an action list defined	Project Team action list?
Change Mgt	Change Management	Change Management
	Define the change management process and practises - follow the current, defined practises (scope, costs, re-budgeting, resources)	Change Management - process (scope, costs and re-budgeting, resources). CRs are managed and people involved know how to do.
	Define / Follow the defined Change Request process and practises	How the Change management is organized in the project?
	Issue Management process is defined	Issue Management and Change request practises
	Define the issue log	How issue management is managed?
		Does project have Issue log?
Risk Mgt	Risk Indicators / Risk Management	Risk assessment for the project
	Risk Matrix done for the project	Is the Risk matrix defined?
	Do Mitigation Plan for the risks	Is Mitigation plan part of risk management?
	Priorities and Severities defined	Functionality most important
	Incidents Impact on Budget, Schedule and Quality	Functionality most visible for users Largest Financial impact
SW Dev.	Software Development	Technology selections
	Do Technology selections and agreements	Service provider selections and Agreements
	Do supplier selections and agreements	Supplier Selections and Agreements
	Organize the needed security audits	Security Audits (Core pas, FE, Integrations)
		Additional Security Audits (from third parties)
Go-Live	Go-Live activities	Go-Live activities
	Define the Go-Live Practises	Are the go-live activities planned or in very early phase identified?
Stakeholder	Stakeholder Satisfaction	Stakeholder Satisfaction
	Follow the process	Make sure the questionnaire documentation is available
Com- municat.	Communication	How internal and external communication is organized?
	Communication channels and practises defined within the project and with share and stakeholders	What are the communication channels? Do you have a communication plan?
	Define the communication Plan and Matrix	How communication has been organized within the project?
	Lessons Learned Practises defined	How Lessons Learned are planned to be done?
Support	Executive support:	Project Support
	compete priorities across various stakeholders (conflicting priorities)	

Project Tools	Project Tools	Project Tools
	Define the project tools - check that tools are in place	
Meeting Practices	Meeting Practices	Meeting Practices
	Meeting Practises definid	
	SG, Project Management, project and development teams, meeting practices	
Documenta-tion	Documentation and Deliverables	Documentation
	Make sure that proper project documentation is done	Is the project documentation clearly structured?
	Documentation repository is defined.	What is documentation repository?
	Project Tools and templates defined for the different level docu-mentation	
Tech.	Technical choices, implementation	Technical choices, implementation
	Tools and technologies are defined.	What tools and technologies are being used?
	Industrial Best Pactises is known and selections are based on those.	Why were these tools and technologies selected?
	Technical decision practises and process defined.	Is the selection in line industry best practices?
	Check that process for making technical decision is in place.	How the technical decisions are made?
	Make sure that technical solution meets non-functional and quality requirements.	How it is planned that technical solution meets non-functional and quality requirements?
	Process to turn functional requirements into the solution defined.	How functional requirements turn into the solution?
	Define / Check that proper version and configuration management	Is there a proper version and configuration management?
	Process for planning the future needs defined. Check	How the planning for future needs will be organized? What aspects have been consid-ered in the planning?
		How the changes are controlled in consistent way?
System Architecture	System Architecture	System Architecture
		How is ensured that pieces that make the solution / product fit together?