

Chithra Prabha Peachi Muthu

Visibility of the project status, usage of agile methods and tools

Microsoft Mobile Oy

Helsinki Metropolia University of Applied Sciences

Degree: Masters in Engineering

Degree Programme: Masters in Business Informatics

Thesis: Visibility of the project status, usage of agile methods and tools

Date: 06 October 2014

PREFACE

The research started in 2012, to study current business needs of the company and how the Business management tools and methods are helping to address the needs, the research began with a study about the best practices and current practices in using agile methods and tools in different teams. The main objective of the research was to analyse the current practices in the teams for project planning and execution and identify the ways that are needed to make the visibility of the project situation clear to all the stakeholders in the project and with a purpose to reduce delays in the project.

While the research moved on with the study and discussion with many individuals in the team, it turned up to be a great opportunity to learn the business management methods and tools from the information shared in the discussions. Also with several years of experience in the industry, in different kind of roles, I had an opportunity to express and relate the best ways of using the tools and methods, which would help any individual in software development organisations similar to the one in the case study.

The research should give a fairly good view of best practices in Agile methods and tools, and what is missing currently in practices, correcting the missing practices could reduce delay in meeting the project deadline. It brings in a clear picture of combining the latest agile methods and tools, to address the current needs of the industry. The solution provided will be helpful for better utilisation of tools and stay updated with the current situation of the projects at any time.

This research would have not been possible without many individuals, whom I wish to give my sincere thanks, and to mention, my mother who travelled to Finland from India on time, when I needed her help to go for studies and my supervisors Jukka Kainulainen and Thomas Rohweder.

Espoo, October 06, 2014

Chithra Prabha

Author(s) Title	Chithra Prabha Peachi Muthu Visibility of Project Status , Usage of Agile Methods and Tools
Number of Pages Date	60 pages + 10 appendices 06 October 2014
Degree	Master's Degree in Engineering
Degree Programme	Master's Degree in Business Informatics
Instructor(s)	Thomas Rohweder ,DSc (Econ) / Principal Lecturer
<p>The aim of the present thesis was to study the current ways of using agile methods and practices in the case company and find the ways to improve the current practices, in order to have better visibility of the project status. Accept360 is the tool used for project planning and execution in the case company, hence the study concentrated on understanding the current practices in using the tool and finding out the gaps that exists in the current practices, in comparison with the best practices of using the Agile methods and tools. Agile approaches are used in software development to help businesses respond to unpredictability. To address the uncertainties and have good visibility of the project status, the agile methods provide rules and a standard ways of usage, which would help the project stakeholders to stay updated with the project status. In the research the best practices in agile method of software development were studied first, and some of the best practices and methods that would best suit the organisation were identified. Next, the current practices were studied in the selected teams in the case organisation, by conducting contact interviews with the key stakeholders, from which the researcher collected information about the current ways of using the agile methods and tools. The interviews also gave information about the opinions about the tool which is being used currently for project planning and execution, how the tool addressed the current needs for project planning and execution and what were the suggestions from the key stakeholders for improving the practices in the tool and methods. The data collected about the current practices was then compared with the best practices and the gaps between them were identified, from which the proposals to improve the visibility of the project situation were created.</p> <p>The study resulted in three key theme proposals. The first theme is to follow the already agreed practices and guidelines, which is missing currently. The second theme is using the Agile Software Product Management method, and the third theme is implementing</p>	

Scrumban practice for project planning and execution. The last two themes mainly recommended ways for having a separation between requirement and engineering management. The proposals were found to be very useful input for further process improvements, to have better visibility of the project status and address the current key issues in the case organisation.

Keywords

Agile Methodologies ,Scrum, Kanban ,Scrumban, Agile Product Management

Table of Contents

1	Introduction	4
1.1	Description of the Company	4
1.2	Business Problem, Objective and Outcome	4
2	Research Approach, Data Collection and Analysis Methods	5
2.1	Research Process Description and Flowchart	5
3	Best Practices in Agile Software Development	8
3.1	Agile Software Development	8
3.2	Agile Methods and Aspects	10
3.3	Scrum	12
3.3.1	Scrum Overview	12
3.3.2	Scrum Work Items	16
3.3.3	Scrum Project Estimation	17
3.3.4	Scrum Project Reporting	18
3.4	Kanban	20
3.4.1	Kanban Overview	20
3.4.2	Kanban Work Items and Terms	24
3.4.3	Kanban Execution	27
3.4.4	Kanban Reporting	28
3.5	Hybrid Project Management Approach	29
3.6	Agile Maturity Model	30
3.7	Agile Software Product Management	32
3.8	Accept360	33
3.8.1	Accept360 Elements	33
3.8.2	Accept360 Team Element Ranking Tab	34
3.8.3	Accept360 Recommended Practices	34
3.8.4	Accept360 Responsible Actors	35
3.8.5	Accept360 Agile Task board for Scrum features	35
3.8.6	Accept360 Kanban features	37
3.8.7	Accept360 Nzilla Support	37

3.9	San Diego Team Practices	37
3.10	Making Most of Scrum and Kanban	38
3.11	Scrumban	39
3.12	Summary of Best Practices	42
4	Current State Analysis	43
4.1	Analysis of teams in Espoo	43
4.1.1	Analysis in Team X	43
4.1.2	Analysis in Team Y	44
4.2	Summary of gaps between best practices and current practices	45
5	Proposal	46
5.1	Theme Proposals	46
5.1.1	Usage of Structured Contents for Work Items	47
5.1.2	Usage of Agile Software Product Management	49
5.1.3	Usage of Scrumban	50
5.2	Summary of Proposal	54
6	Conclusions	55
6.1	Summary	55
6.2	Practical Implications	55
6.3	Evaluation	56
6.3.1	Objective Vs Outcome	56
6.3.2	Reliability	57
6.3.3	Validity	58

Abbreviations

BL Backlogs

PM Project Manager

UX User Experience

UI User Interface

QA Quality Assurance

A360 Accept360 tool

NFT Non Functional Testing
 FT Functional Testing
 LVT Language Variant Testing
 MMF Minimum Marketable Feature

LIST OF FIGURES

Figure 1 Research Execution Flowchart	7
Figure 2 Scrum Skeleton	13
Figure 3 Scrum Recommendations.....	14
Figure 4 Estimation Graph	17
Figure 5 Burndown Chart.....	20
Figure 6 Kanban Implementation Model	21
Figure 7 Kanban Value Stream.....	24
Figure 8 Kanban Work Items	24
Figure 9 Kanban Board.....	27
Figure 10 Kanban Cumulative Flow Chart.....	28
Figure 11 Hybrid Project Management Skeleton.....	30
Figure 12 Agile Maturity Model Levels	32
Figure 13 Agile Software Product Management.....	33
Figure 14 Accept360 Structure in Practice	34
Figure 15 Accept360 Actors.....	35
Figure 16 Scrumban Sprint and Value Stream.....	40
Figure 17 Template for Work Items in Scrum team	49
Figure 18 Sample Framework with Scrumban Method.....	52
Figure 19 Sample Scrumban Board.....	53

1 Introduction

1.1 Description of the Company

Earlier Nokia Oy comprised following business groups: Smart Devices, Mobile Phones, Locations and Commerce, and Nokia Siemens Networks. Later Nokia's Smart device and Mobile Phones groups were then bought by Microsoft and merged into one, to create a new group Microsoft Mobile Oy, ever since the merger, over the past three years the group has created incredible results: award-winning phones and amazing services that have made Nokia Windows Phones the fastest-growing smartphones in the world. In Microsoft Mobile Oy, Applications Software teams which belong to Smart Devices unit are the teams which creates valuable and innovative solutions that adds value to the smartphones.

1.2 Business Problem, Objective and Outcome

Projects executed by the teams in Applications Software Team require dynamic changes and updates in requirements always. Projects are executed in challenging situations where the market is highly competitive and customer needs are changing rapidly, in order to meet such demands and supply the best products for the customers ,and also to deliver products with high standard and quality, it is required to make sure that the projects are planned and executed with high productivity and good utilisation of the resources available, also it is required that the project situation is visible to all the audience of the project at any point of time, so that the needs for executing the project successfully are meet as early as possible and there is minimum delay in delivering the project. Even after having a well-defined processes and tools for project planning, management and execution, there are delays in some projects. Hence there is a good need for evaluation of how the project planning tool and methods that are used in the projects and how efficiently the tools are used to track the progress of the project.

The main objective of this thesis is to study the best practices using the agile methods and tools, in the company Accept360 is being used mainly for project planning and tracking, the current way of usage of Accept360 tool need to be analyzed, and the gaps that

exist between best practices and the current practices need to be identified, and solutions that could fill this gaps need to be researched.

Research question of this thesis is “How the visibility of the project situation could be kept updated at any point of time?” “What are the best ways to stay on track of the current situation of the projects and so be able to manage the issues appearing earlier and minimize the delivery slippages?”

The outcome of the study is to propose the solutions which include guidelines for the efficient usage of best practices, artefacts, tools and methods, sample process framework that would address the needs of the projects and that are needed to help everyone involved in the project to have a clear visibility of the project status.

2 Research Approach, Data Collection and Analysis Methods

2.1 Research Process Description and Flowchart

This chapter explains research approach, this includes the steps that were done to identify the research problem, analyse the current state and the steps that were done to find solution for the research problem.

As first step, the details about the case company was studied as a part of the course on Business management methods and tools, it included complete study of the company’s strategy, mission, visions, goals, current growth needs and KPI measurements .In the study it was found that the current importance for the company is to minimise the delays in delivering the software, for which the visibility of the project status plays a key role, unless the needs of the project is visible to all the stack holders involved , at any point of time, it is not possible to address them earlier. In ASW team Accept360 tool is used as the key tool for tracking and executing the projects, hence a research on how the tool is currently being used in the teams, does the tools address all the needs of the projects, is there any improvements needed in currents way of working in the teams needed to have better visibility of the project situation sounded a good topic of research for the thesis. Hence a study about the Visibility of the project situation was decided as topic of the research. A flowchart in Figure 1 was created with steps for finding the solution for research problem.

As second steps, the research topic was discussed with the senior manager in the company, and it was considered that this topic of research would yield a good review and guide for identifying the gaps that exists in the currents practices of usage of the tool and methods for Project Planning and Execution, and the objective of the thesis was set. Since Project planning and tracking is done using the Accept360 tool ,study of the tool was planned first , Initially couple of meetings were planned to discuss the available information about Accept360 tool and how the research need to be done. During the first meeting the guidelines available for the Accept360 tool were discussed, and previous presentations that were used to provide guidelines for the teams, for using the Accept360 tools were gone through. In the second meeting a short walk through of contents in the Accept360, for some of the projects were viewed and discussed. Then identification of the teams and the group of people to be contacted for current state analysis was done. Next identification of a team that is using Accept360 in compliance to the guidelines was done, it was found that the Team S, in Sandiego, has created the content in Accept360 in a best format, that it could be used as benchmark for comparative study of Accept360 content.

As a third step, a study of best practices in the software development methods and practices was done using books, articles, company's project guidelines and the training materials for the Accetp360 tool and practices. Next access to the training website for the Accept360 tool, which is an exact trial ground for the usage of the tool in the same way as real situation was requested, with access to the tool, studied the features supported in the tool, experimented the usage of the tool to understand the complexities and merits of the tool. Next a basic set of questions and topics for discussion were designed for the interview.

As a fourth step, contact interviews were done in different teams and in each team people from different kind of roles were contacted and discussion about the usage of the tool and about the team practices was done.

As a fifth step, the gaps between the best practices that were studied from the second and third step were compared with data collected in the fourth step.

As sixth step, a proposal of the solutions to have better visibility of project status and minimize the delay using the tools and practices were created and discussed with the research supervisor in the company.

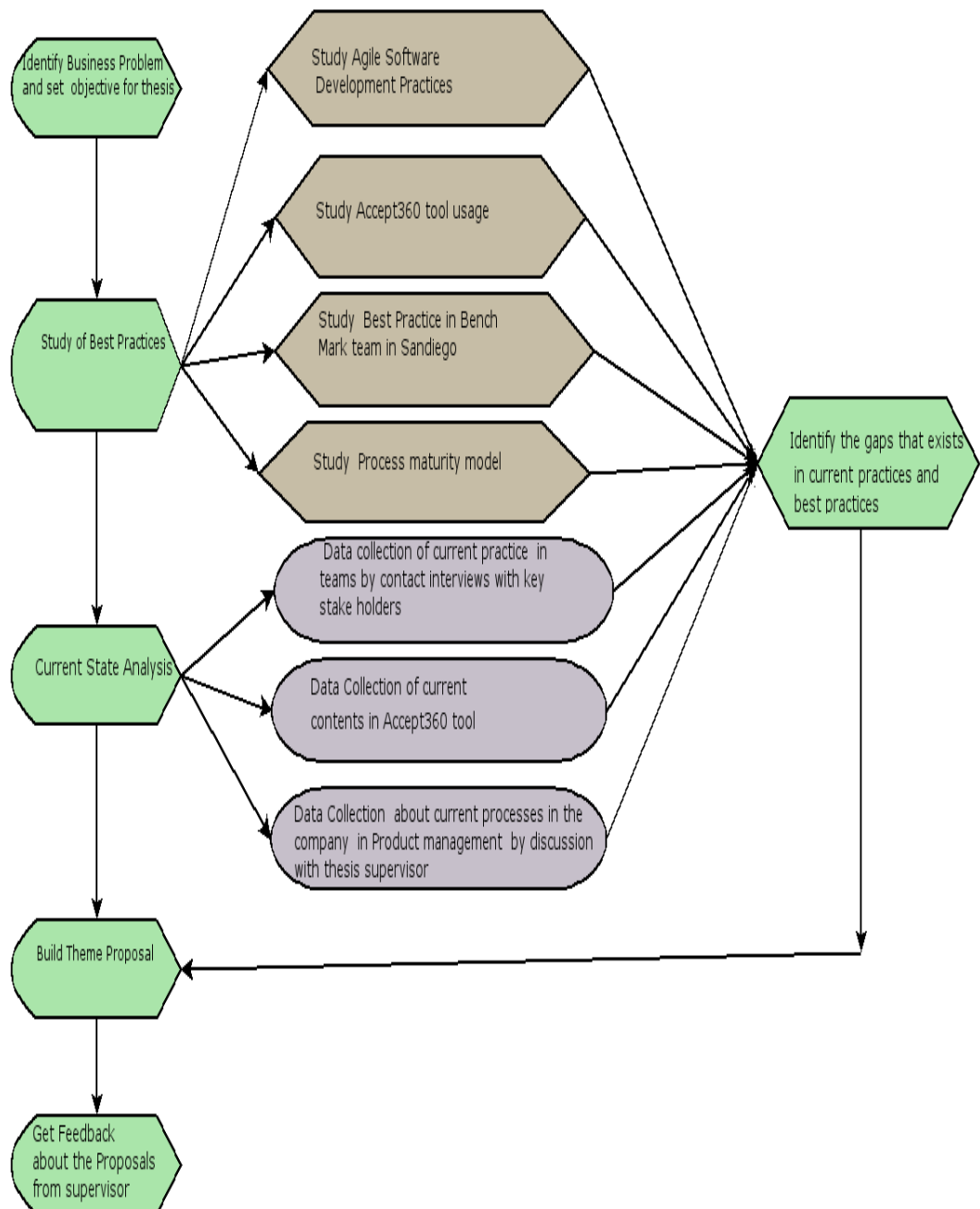


Figure 1 Research Execution Flowchart

3 Best Practices in Agile Software Development

This chapter includes the details of the data studied for understanding the best practices in Agile software development methods, best practices in using the Accept360 tool, the Project Management Methods and Process Maturity Model.

Firstly, the Best practices of Agile Software Development methods were studied from a list of the books and articles prepared on the search topics of the following keywords Agile software development methods, scrum, kanban development methods.

Secondly, Best Practice of using the Accept360 was collected from guidelines, presentations and tutorials available for the Accept360 tool in the intranet and discussion with the supervisor.

Thirdly, the practices followed in the San-diego team was discussed in a call and the practices were noted in the discussion notes, this include the walk through of the contents the team has in the Accept360 tool.

Forthly, a study on the project management methods and techniques were studied and it was identified that the Hybrid Project Management approach and Agile Product management methods are the two different methods that are used and that would suit well for the software development organization in case study, hence a details study and discussion about this methods were discussed with the supervisor.

Lastly, the key findings in the best practices were summarised, so that the data from the study could be used for current state analysis.

3.1 Agile Software Development

From the work experience in Nokia Research and Development centre since 2006, Nokia research and development organisation until and around 2007 had been following the waterfall model for its software development process, after that time there were many trainings and changes in the way of working, for moving towards Agile

methods. The main drawback with waterfall model is, it follows the procedure in which when each stage of the project is completed, the developers go on to the next stage, and they do not validate the previous step when they go forward, so at the end if the final product turns out to be with some flaws, there is less chance to go back and fix the issues, if the flaws were found at the end, to go back and fix the problems means it would be possible only by scratching the whole project, there is no chance for change, hence extensive planning on expected outcome is done at the initial stage and then the project is executed, after the planning is done the design and requirements are documented very well, the documentation helps when there are very less resources available to continue in the project, help to add new resources quickly. In waterfall model client would know what to expect, but at the same time there is a high risk that if the planning at the beginning had some faults, that could end up in risking the whole project and it also does not consider the evolving needs of the customers, testing is done only at the end of the project, so if errors are found at this stage, it could be that they will stay for rest of the project and could not be fixed. Waterfall method is good only if the initial requirements are very clear and expected outcome is well defined.

Agile methods were introduced as solution to many problems in the waterfall methods, this method allows changes to be done at any stage of the project, initial requirements are kept very simple and a very simple product is made initially, later changes are done as per the client needs and latest improvements in the industry, The products development is done in small cycles in stages, at the end of each cycle the requirements are prioritized and updated, testing is done at every sprint, so that the errors are identified at every stage and the prioritised errors are taken care in next development cycle, this also produces good quality product, over comprehensive documentation at every stage, this means that the product is in quality at each stage and could be released at the end of any stage, this ensure that the deadline for releasing the product is always met. At the same time there is more care needed, to make sure that every stages in the project are not series of coding stage and end up in a way that the initial planned requirements and product and final outcome are totally different. Agile method of development is good if the product is intended for a rapidly changing market. This also requires highly skilled, independent and adaptive developers to be involved in the project.

3.2 Agile Methods and Aspects

Agile methodologies promise that the product reaches the market faster, with good quality of software fulfilling customer needs. There are several software development methods introduced based on the principles that are followed in different organizations. For example the Lean Software Development method was introduced based on the principles followed in Toyota manufacturing unit, which are based on continuous improvement and elimination of waste, but these are more generic principles, that this could also be applied in the agile methods. The table below from the reference [1], lists different agile methods and principles mentioned in the reference [1]

Agile Software Development Methods and Principles	
DSDM or The dynamic systems development methodology	DSDM has three broad phases that consist of requirements prototypes, design prototypes, and then an implementation or production phase (Millington & Stapleton, 1995).
Scrum	Scrum is a light weight software development process consisting of implementing a small number of customer requirements in two to four week sprint cycles (Schwaber, 1995).
Extreme programming or XP	XP consists of collecting informal requirements from on-site customers, organizing teams of pair programmers, developing simple designs, conducting rigorous unit testing, and delivering small and simple software packages in short two-week intervals (Anderson et al., 1998).
Crystal	Crystal methods involve frequent delivery; reflective improvement; close communication; personal safety; focus; easy access to expert users; and a technical environment with automated testing, configuration

	management, and frequent integration (Cockburn, 2002a).
Feature driven development or FDD	FDD involves developing an overall model, building a features list, planning by feature, designing by feature, and building by feature (Palmer & Felsing, 2002).
The rational unified process or RUP	RUP involves a project management, business modeling, requirements, analysis and design, implementation, test, configuration management, environment, and deployment workflow (Kruchten, 2000).
Adaptive software development or ASD	Adaptive software development or ASD involves product initiation, adaptive cycle planning, concurrent feature development, quality review, and final quality assurance and release (Highsmith, 2000).
Lean Software development or LEAN	LEAN involves eliminating waste, amplifying learning, deciding as late as possible, delivering as fast as possible, empowering the team, building integrity in, and seeing the whole (Poppendieck & Poppendieck, 2003).

In general, agile methods may vary in applying certain practices. However, the methods emphasize on producing working software in small iterations, and utilising resources efficiently.

Main features of the agile software development could be summarized as three key things, as mentioned in the reference [4], Feature orientation, Reactive development and Evolving Project scope. Feature orientations is to have main focus on the producing features faster, with a goal to deliver the working functionality that brings most value to the customer. Reactive development is reacting to a change rather than planning ahead and keeping the decision making as late as possible. Examples of reactive practices are re-

factoring, adjusting requirements priorities, and release scope after each iteration. Evolving project (release) scope is the main distinguishing features of agile development, changing from a fix-scope approach to a more open-ended approach. In the traditional fixed scope approach much effort is spent on defining and planning the content of a product release of a project upfront. In agile the release scope is emerging in the process of development rather than planned ahead. A prioritised list of requirements serves as an initial input which is a kind of wish-list of a release scope. The release scope is expected to be refined and updated at the end of each planned cycles of work, in order to accommodate changes and new information that was learned in the latest stage. This practice is in line with the reactive development property described above.

3.3 Scrum

3.3.1 Scrum Overview

Scrum's main principle is implementing a small number of requirements in a short cycle. It includes mainly the following ways, Self-correction with inspection, making everything visible or known to the stakeholders, for example, plans, schedules, issues and progress more clear at every point of time as they are changing all the time, Stop and Review the product and the process.

Scrum's skeleton is shown in Figure 2. It includes the Daily Scrum, Sprints and Increments. Sprints contain the iteration of activities that occur one after another. The output of each sprint is an increment of product, which could be released to the customers. During the sprint, every day the individual team members meet to inspect each other's activities and make appropriate adaptations. Driving the sprint is a list of requirements. This cycle repeats until the project is no longer resourced for execution.

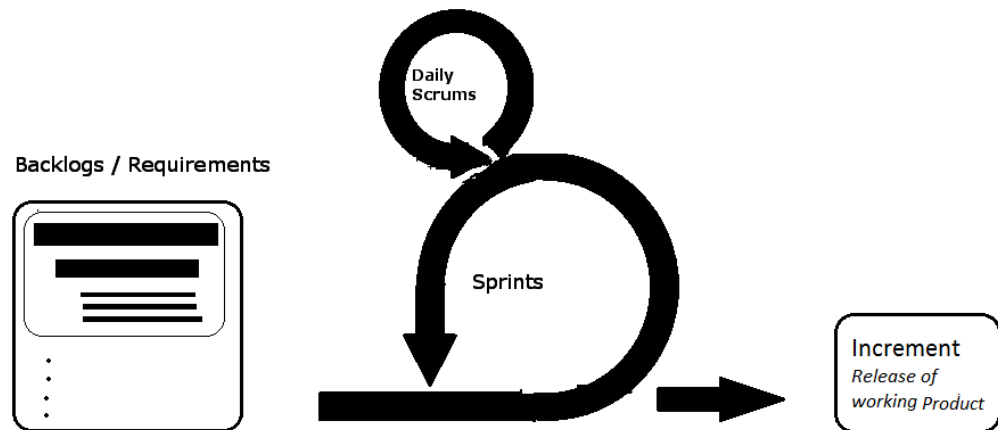


Figure 2 Scrum Skeleton

Scrum operates this way: At the beginning of each sprint the team reviews what it should do. Then the team selects what it can turn into the release of a potentially working functionality by the end of the sprint. Then the team start to work independently with no interruption to make the best efforts for the rest of the sprint. At the end of each sprint the Team demonstrates the release of functionality it built, so that all the stakeholders can inspect and do adaptation to the project.

In Scrum the project is started with a vision that needs to be developed. The vision initially is stated in market terms and not much in technical terms. The items which are to be developed to deliver this vision are made as a list called product backlog items. As scrum suggests one person is responsible for delivering the items planned to those who are funding the project and the person in scrum terms is called Product owner. The product owner is responsible for delivering the vision in such a way that maximises their return of investment, for that the owner creates a plan, which includes the product backlog.

Product Backlog is a list of functional and non-functional requirements, that when turned into functionality, will deliver the vision of the project. The requirements that are important to create most value for the product is at the top priority in the list of requirements. The prioritized requirements is a starting point, and the contents, priorities, and grouping of the requirements into releases, this is usually expected to change the moment the project starts. The updates and changes in business requirements and the performance of the team to build the requirements into functionality decides the changes in the requirement list and its content.

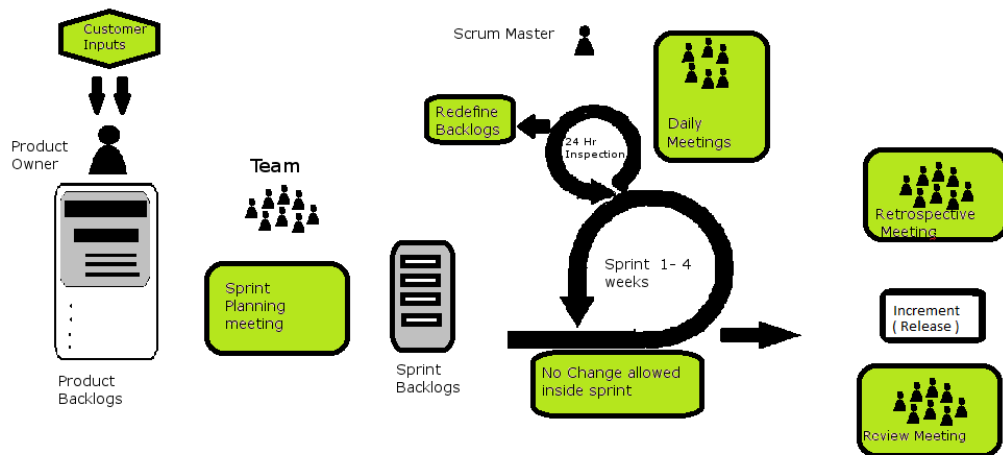


Figure 3 Scrum Recommendations

Scrum team is a self-organizing team. The team plan their own work, the work plan is visible as sprint backlog items, it is a good practice if each functionality to be delivered is appearing as a sprint backlog item, this helps the team to view the sprint backlog item as a plan and also as a reference for the other team members as they work. In this way it ensure that the work to be done is thought through. This would make sure that the daily scrums are more meaningful as the work plan is clearer.

Scrum master role is recommended in Scrum, one person in the team act in this role, to ensure that the scrum rules are followed and the team do not cut corners. Scrum recommends it as a thumb rule that the team is given the prioritised backlog items, based on

which the team decide by its own on what need to be done next, and it is required that the scrum master in the team ensures that the team does not skip the process of scrum, and also can help remove the impediments, but at the same time the person does not have the authority over the team, but only acts to help shape the development processes in the team and make sure that the team brings out the good results and make sure the team is not going off track.

Scrum team location is recommended to be in a collocated team space, which is achieved by removing the cubicles, and let team members to face each other and communicate often, this eliminates isolation and misunderstandings.

Mandatory meetings for scrum execution are, the Sprint planning meeting, Daily Scrum meeting, Sprint review meeting and Sprint retrospective meeting, these meetings are supported and encouraged by the Scrum master, and these meetings are emphasised to get the maximum benefit of the scrum process framework. In Scrum all work are done in small cycles called sprint, which usually is a one to four weeks cycle.

Sprint Planning meetings usually have two parts, in the first half the Product Owner presents the team with the list of requirements that are prioritised and that are expected to deliver a functionality in the release at the end of the Sprint, this list is called Sprint Backlogs. During the second half of the meeting, the team is expected to discuss and plan their work, tasks are created by the team for completing the Sprint backlog items, after this meeting the sprint is started, and it is time boxed to the number of days the sprint is supposed to be.

Daily Scrum meeting is held daily for 15 minutes, in this meeting the all the team members meet at one place and each team member answers to the three questions, what was done since last daily scrum? What is planned to be done from now till the next daily scrum? And what are the impediments that are on the way to complete the planned items in the sprint backlog. The purpose of this meeting is to synchronize the work with everyone in the team and to address and schedule for more meetings if needed to go forward in completing the sprint goals.

Sprint review and the Retrospective meetings are held at the end of each sprint. In the sprint review meeting the team presents the developed functionality or implementation to all the stakeholders of the project, this is an informal meetings, this helps to bring all the stakeholders together and have an understanding and view what is needed next for the project and then plan for it.

Sprint retrospective meeting is held after the Sprint review meeting and before the next planning meeting. This meeting is to encourage the team to revise the process they adapted in the sprint, and make sure it is within the scrum process framework, so that the next sprint is more effective and interesting for everyone in the team.

3.3.2 Scrum Work Items

User Stories, Epics, and Themes are the three work items which are mainly used to define the items that needs to be worked on by the scrum team. Theme in scrum is the highest level in the story hierarchy and describes a view of a tangible product which can be a trading application or an abstract goal such as performance tuning. A product owner breaks down a theme into one or more epics.

Epic represents a group of related user stories or a block of requirement that is not yet been rationalized into stories. Sometimes a large user story is also called as an epic.

A story is a brief statement of a product requirement or a business case. Typically, stories are expressed in plain language to help the reader understand what the software should accomplish. Product owners create stories. A scrum user then divides the stories into one or more scrum tasks.

Scrum tasks are the discreet pieces of work required to complete a story.

Scrum recommends to remove the unneeded artefact such a design documents, hence the work item's descriptions serve as a source of information about the project.

3.3.3 Scrum Project Estimation

Projects are estimated based on the three key things, the resources, scope and time. Estimation is a challenging task to do when there are uncertainties of way of execution and process followed are changing, but when there is a defined process it could make estimation easier. Time spend for estimation could go waste if the estimation are not useful or appropriate, but the possibility to have accurate estimate is very less, as there is some amount of uncertainties in the projects dependencies, in projects which are driven for innovation and change of existing technologies, so spending more time in estimating the project will not return a good value for investment, but spending a little time to estimate even if it is less accurate would be more efficient to manage the project.

The curve below give a picture of dependency of accuracy in estimates and efforts spend in estimation.

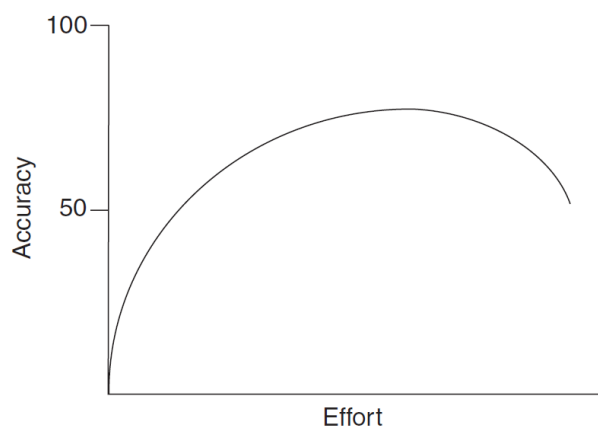


Figure 4 Estimation Graph

From the above Figure 4 indicates that the accuracy value starts to degrade when the amount of time spend in estimation is going beyond certain limit. Estimate is still an estimate. Spending more time in estimation is not going to make the estimate more accurate. And to reach high level of accuracy in the curve, which means to move away from base line ,only a little efforts is needed, so a less amount of time spend in the estimation would increase the accuracy level.

As mentioned in the reference Agile teams tend to stay in the left side of the Figure 4, as they acknowledge that estimates could not be done accurately, but still encourage the idea of estimation with less time and recognise those estimates that give big gain. As agile process is to deliver frequently a fully working, tested and integrated software, they always are in a situation that they have a reliable plan.

Usually the user stories are not fully grained down to fine level, so they are not of same order of magnitude, and differ in their size, Hence by aggregating some stories into themes and writing some stories as epics, a team is able to reduce the effort they will spend on estimating. However, it's important that they realize that estimates of themes and epics will be more uncertain than estimates of the more specific, smaller user stories. User stories that will be worked on in the near future, for example in the next few iterations, need to be small enough that they can be completed in a single iteration. These items should be estimated within one order of magnitude. It is a good practice to use the sequence 1, 2, 3, 5, and 8 for this estimation.

User stories or other items that are likely to be clearer than a few iterations can be left as epics or themes. These items can be estimated in units beyond the 1 to 8 range sequence recommend earlier. To accommodate estimating these larger items it is good to add 13, 20, 40, and 100 to the preferred sequence of 1, 2, 3, 5, and 8.

3.3.4 Scrum Project Reporting

In Scrum project there are four reports that need to be created by the end of each sprint. The first lists the Product Backlog at the start of the previous Sprint. The second lists the Product Backlog at the start of the new Sprint. The third, the Changes report, details all of the differences between the Product Backlogs in the first two reports. The fourth report is the Product Backlog Burn down report.

The Changes report summarizes what happened during the Sprint, what was seen at the Sprint review, and what adaptations have been made to the project in response to the inspection at the Sprint review. Why have future Sprints been reformulated? Why

was the release date or content reformulated? Why did the team complete fewer requirements than anticipated during the Sprint? Where was the incomplete work reprioritized in the Product Backlog? Why was the team less or more productive than it had anticipated? All of these questions are answered in the Changes report. The old and new Product Backlog reports are snapshots of the project between two Sprints. The Changes report documents these differences and their causes. A collection of Changes reports over a period of time documents the changes, inspections, and adaptations made during that period of time.

Burn down Report: This Burn down report measures the amount of remaining Product Backlog work in the units of story points for each of them, on the vertical axis and the time scale, by Sprint days, on the horizontal axis. A Story Point is a subjective unit of estimation to estimate User Stories. It represent the amount of effort required to implement a user story. Using Story Points for estimation is better than estimating in hours or days, as it is an estimation done using relative sizing, by comparing one story with a sample set of previously sized stories. Relative sizing across stories tends to be much more accurate over a larger sample, than trying to estimate each individual story for the effort involved. The Fibonacci series (1, 2, 3, 5, and 8) is most commonly preferred to categorize efforts in scale of 1, 2, 4, 8, 16 points and so on. The Product Owner plots remaining quantity of Product Backlog work at the start of each Sprint. By drawing a line connecting the plots from all completed Sprints, a trend line indicating progress in completing all work can be drawn. By figuring out the average slope over the last several Sprints and velocity can be determined, occurring when the trend line intersects the horizontal axis. This is an important report. It would graphically present to management how the factors of functionality and time were interrelated. This is good to be included in the project reports but can be an appendix

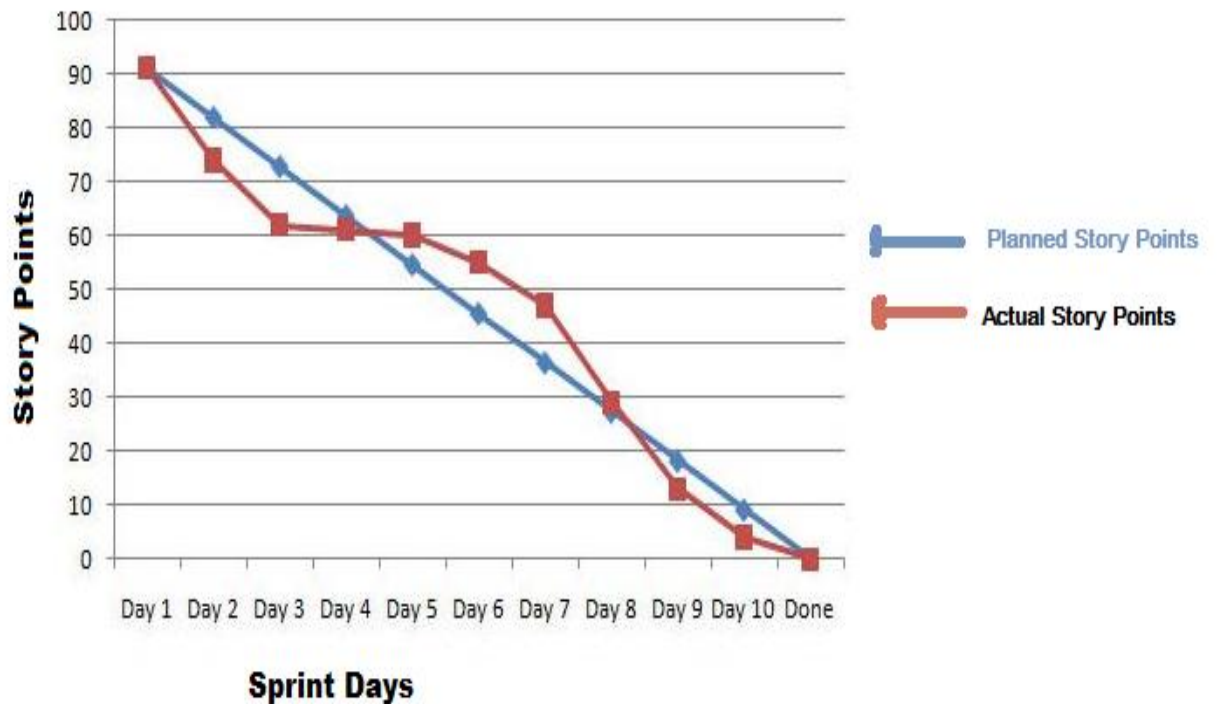


Figure 5 Burndown Chart

3.4 Kanban

3.4.1 Kanban Overview

Kanban was initially designed and used in manufacturing units and then it had its way in to many other fields as well and in software development too, the basic implementation model of Kanban described in the reference [7] gives a clear idea of steps needed to a Kanban in any field. Kanban itself is a scheduling tool, which replaces the traditional daily and weekly scheduling. Kanban scheduling is an execution tool rather than planning tool. Kanban does not replace the planning process, but rather takes the information from planning and uses it to create the Kanban, a Value stream which is explained later in this section. Kanban is considered as a tool which helps to identify and remove the project dysfunction. The word Kanban is translated as 'visual cards'.

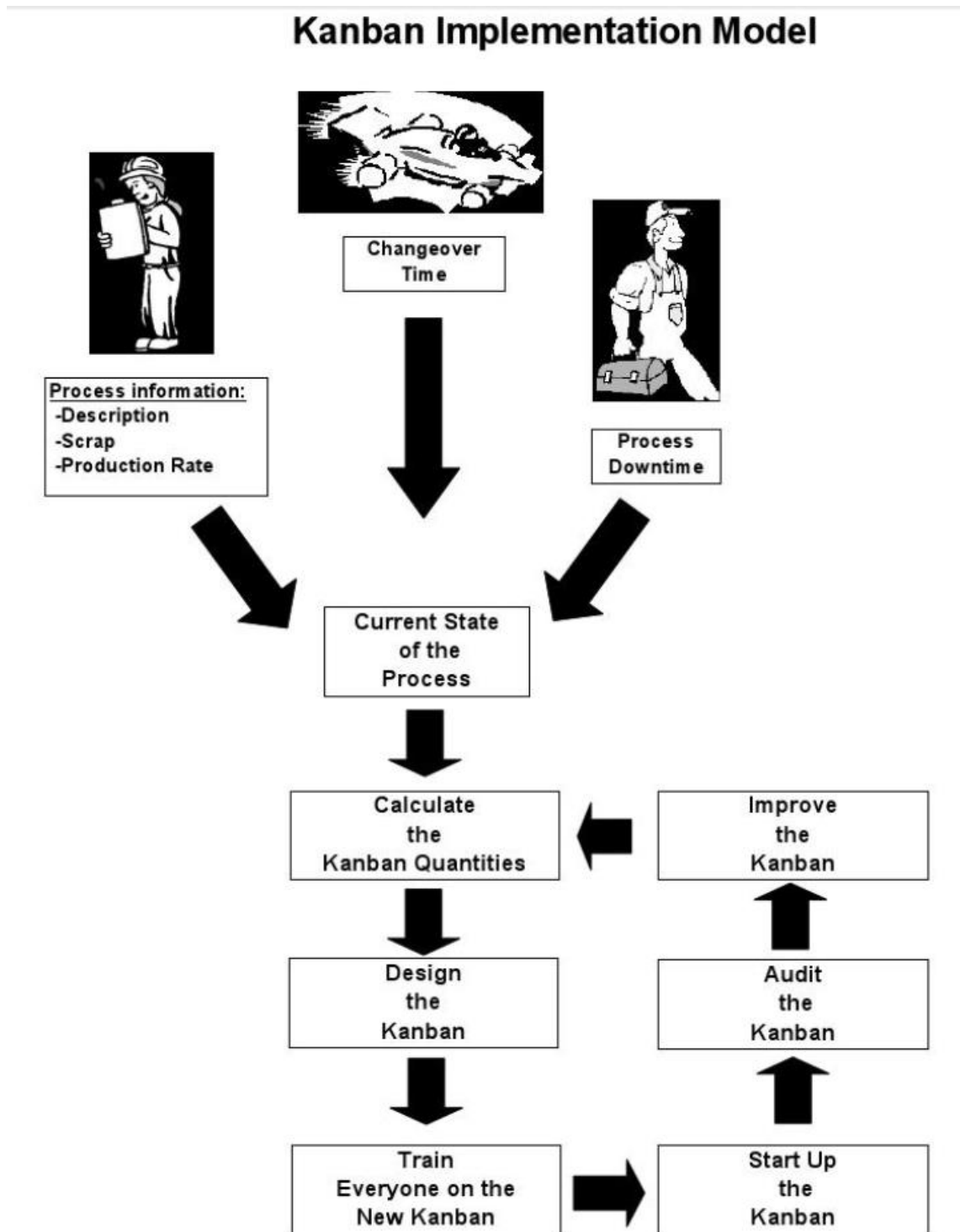


Figure 6 Kanban Implementation Model

Kanban model above has the seven steps for implementing Kanban for a production organisation, but the same steps would apply for any organisation, like software development units. The steps are 1. Conduct data collection 2. Calculate the Kanban size 3.

Design the Kanban 4. Train everyone 5. Start the Kanban 6. Audit and maintain the Kanban 7. Improve the Kanban.

Step 1: Conduct Data Collection: This Phase is to collect the data necessary to characterize the development process. This is conducting value stream mapping (VSM) for the entire organisation, which is to determine which development processes would be good candidates for implementing pilot Kanban scheduling systems.

Step 2: Calculate the Kanban Size: This step is to calculate the size of the Kanban. Initially, calculate the Kanban work item size based on current conditions, not based on future plans or desires. The initial calculations will utilize the development requirements, the productivity rate and risks involved.

Step 3: Design the Kanban: This step is designing the stages and flow in the Kanban, the value stream, Once the Kanban quantities required to support development requirements based on current conditions is calculated ,it is time to design the Kanban, The completed Kanban design will answer the question of how you will implement the Kanban. The design will consider:

- How will the work items be controlled?
- What are the visual signals?
- What will be the rules for conducting the Kanban?
- Who will handle the Kanban transactions?
- Who will make the scheduling decisions?
- Who will resolve problems?
- What visual management items will be needed?
- What training will be required?
- What is the implementation schedule?

The end product of this step should be a plan for implementation of the Kanban, including implementation actions, action assignments, and schedule milestones.

Step 4: Train Everyone: The people involved has to be trained about how the system will work and on their role in the process, the process and the visual signals has to be explained in a training. Also, the rules are reviewed during the training. It is aimed for taking the participants through what-if scenarios, to help them understand their roles and the decision-making process. The training is focused on operating the Kanban.

Step 5: Start the Kanban: Before Kanban scheduling is implemented, all the visual management pieces are kept in place. To avoid confusion and make training much easier, the signals are set up, control points are marked, and the rules are completed and coordinated. As the Kanban is deployed, it is good to anticipate problems that may impact success and take action to prevent or mitigate these problems. During the deployment stage, develop a scheduling transition plan, determining the exact point for the change and the amount of efforts required to make the change.

Step 6: Audit and Maintain the Kanban: After the Kanban starts, the next step is of the process, auditing the Kanban. When the Kanban is designed the person who will audit it, is also identified. Typically, the auditor will be watching how the scheduling signals are handled and whether output stays satisfactory. When the auditor finds problems, then the problems need to be fixed immediately by the responsible party to maintain the integrity of the Kanban design. The auditor will have to look at future requirements to make sure the Kanban quantities meet expected demand.

Step 7: Improve the Kanban Finally: After the Kanban gets running, look at how to improve the Kanban to reduce new work items waiting to enter the stream. Resist the urge to just start pulling items. Check how the flow is running, and pull the necessary items immediately. After this one-time adjustment, only reduce the quantities based on improvements made to the development process. Determine the amount that can be reduced by using the calculations used in sizing the Kanban to calculate the new quantities.

The main idea of the Kanban is to have a flow of stages, a running value stream which has different level, it is the part of step to design the Kanban, and is shown in the Figure 7 below.

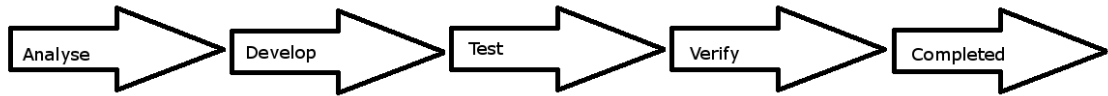


Figure 7 Kanban Value Stream

Kanban works in a way that the items are pulled into each stage of the value stream in a flow. Each stage in the stream has two states: Queue and Execution state. When an item is moved from one stage to another, it first stays in Queue and is then moved to Execution state. And there is a limit set for the total number of items in each stage, this is called work in progress limit, this limit is set based on the capacity of the team, so this makes it a value pulling system to keep the flow steady with work in progress limits in each stage being uniform, when the items in each stage exceed the work in progress limits or if the items in a stage are empty, it is a signal that the flow needs attention, also makes the progress more visible to all stakeholders.

3.4.2 Kanban Work Items and Terms

Most of the agile practitioners use the term iterations, which is a time-boxed cycle in which the selected user stories are completed. In Kanban, which is more specifically designed for Lean practitioners, the term iterations is replaced by Minimum Marketable Features (MMF). In Kanban, the team works on the MMF with no time limits, for each MMF user stories and for each story, the scenarios are created. Each Story is a card that represents functionality and Scenario represents the action related to each functionality.



Figure 8 Kanban Work Items

Delivery Rate: The rate at which units of work (work item) pass through the value stream or part of the stream.

Lead Time the term is Development Delivery Rate, Measured in: "units" per day/hour/second, the two terms described above can be related as

$$\text{Lead Time} = (\text{Work In Progress}) / (\text{Delivery Rate})$$

$$\text{Delivery Rate} = (\text{Work In Progress}) / (\text{Lead Time})$$

Value-adding Time: The total time spent on value-adding activities for one unit of work. Value-adding activities exclude waiting and superfluous work.

Resource Efficiency: A measure of the utilisation of a given resource, i.e. the ratio between the time working on adding value in the system to the total time available.

Flow Efficiency: A measure of time-utilisation on a given unit of work, i.e. the ratio of the Value-adding Time to the (System) Lead Time.

Backlog A non-WIP-limited queue containing work items awaiting service by the initial activity in a Kanban system.

Work Item The item controlled in the Kanban system; Effort Required Determines the approximate size of work in person-units of time. May be a negotiated function of desired quality.

Cadence The rhythm of the production system. Not necessarily an iteration. Kanban still allows for iterations but decouples prioritization, delivery and cycle time to vary naturally according to the domain and its intrinsic costs. The average transit time of a work item through a Kanban system.

Activity: Value-adding work that can be determined as complete. Includes: activity queue, a set of resources, and a WIP Limit. Represents an allocation of the effort required to complete a work item.

Next Work Item Selection, Function Rule for selecting the next work item from a queue when an activity has less work than its WIP limit; depends on both Class of Service and Value Function, and leads to specific flow behaviours.

Class of Service (CoS): Provides a variety of handling options for work items. A CoS may have a corresponding WIP limit for each activity to provide guaranteed access for work of that class. A CoS WIP limit must be less than the activity's overall WIP limit. Examples are expedite, date-certain and normal. CoS may be disruptive (such as expedite) and is the only way to suspend work in progress.

Value Function: Estimates the current value of a work item within a CoS for use in the selection algorithm. Can be simple (null value function would produce FIFO) or a complex, multiple kanban-system, multi-factor method considering shared scarce resources and multiple cost/risk factors. The means of prioritizing work items.

Activity Queue: Holds work items within an Activity that are awaiting processing. The sum of items in process and items in activity queue must be within the WIP limit for each CoS.

WIP Limit: Limit of work items allowed at one time within an activity. Prefer this term to flow units in process or similar. (Measured in: "units".) Which term should be used for the rate at which units pass through the system or part of the system? Velocity (in Scrum), Delivery Rate and Throughput are all used frequently - probably Delivery Rate is more common in the Kanban community, though I have a slight preference for Throughput. It is only one word, and it applies equally to a subset of the system as to the final delivery part.

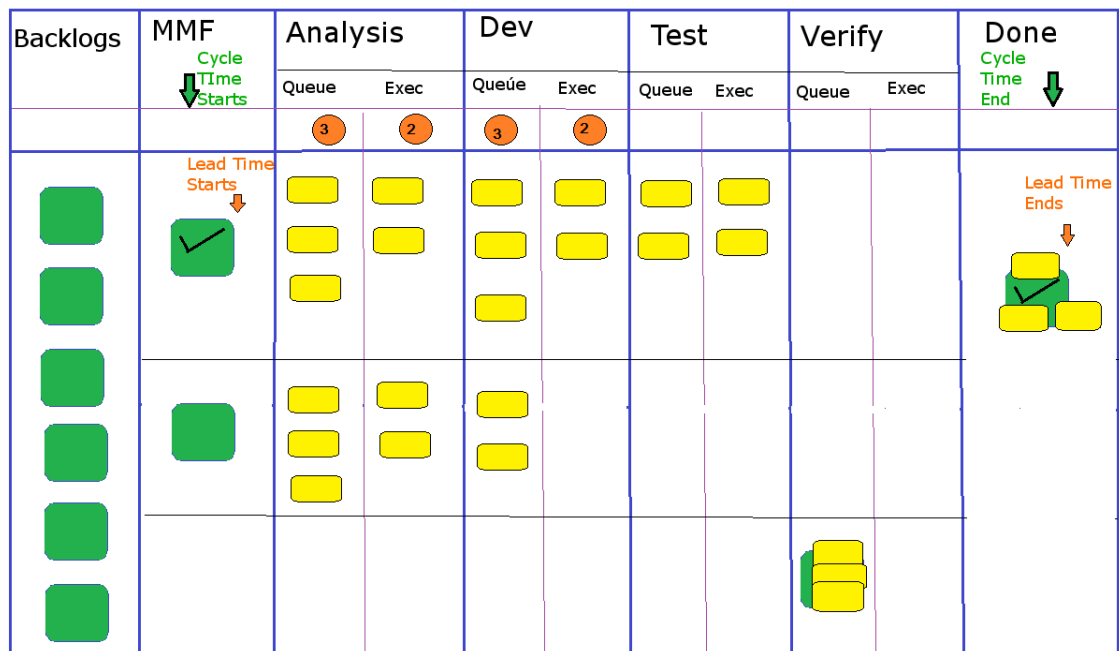
Visible Representation: A common, visual indication of work flow through the activities; Often a columnar display of activities and queues. May be manual or automated. Shows status of all work-in-progress, blocked work, WIP limits it is a characteristic that provides transparency enabling better management. Difficult to model.

Flow Metrics Includes cumulative flow charting and average transit (lead) time.

3.4.3 Kanban Execution

In Kanban work is pulled from the back rather than pushed from the front. Limiting the number of items in any one flow stage at a point of time, with work in progress limit (WIP) is the key factor which drives the Kanban execution, forcing WIP limits encourages, team members to stop at one point when the WIP limit has reached and everyone looks in to the issue together until it is solved before they move on to work on the next item, In this ways the impediments and roadblocks are eliminated as early as possible. Kanban contains an embedded process for handling items that need to be expedited through the flow, fixed delivery dates, and work type splitting

Work is assumed to be broken down to a roughly similar size. In the Kanban Board not only the flow of stories or scenarios are represented, but the MMF itself also could be added in the flow and checked. Prioritisation of the backlog is performed just in time (JIT).



● -- Work In Progress Limits

Figure 9 Kanban Board

3.4.4 Kanban Reporting

The key means to check the progress of work in Kanban team is using Cycle Time, Lead Time and Cumulative Flow chart. Cycle time is the time taken for a unit (in terms of a user story or a scenario) to pass through all the stages in the flow. Cycle time starts when the flow starts and when any one of the unit is out the flow to a completed. The main difference between the cycle time and Lead time is the unit is marked in the later, the time between when on particular unit it entered to the flow and the same item is out of the value stream as complete.

In the Figure10 below is the view of the Cumulative Flow Diagram by Kanban Tool. Coloured areas on the diagram represent work in progress for each stage of a process.

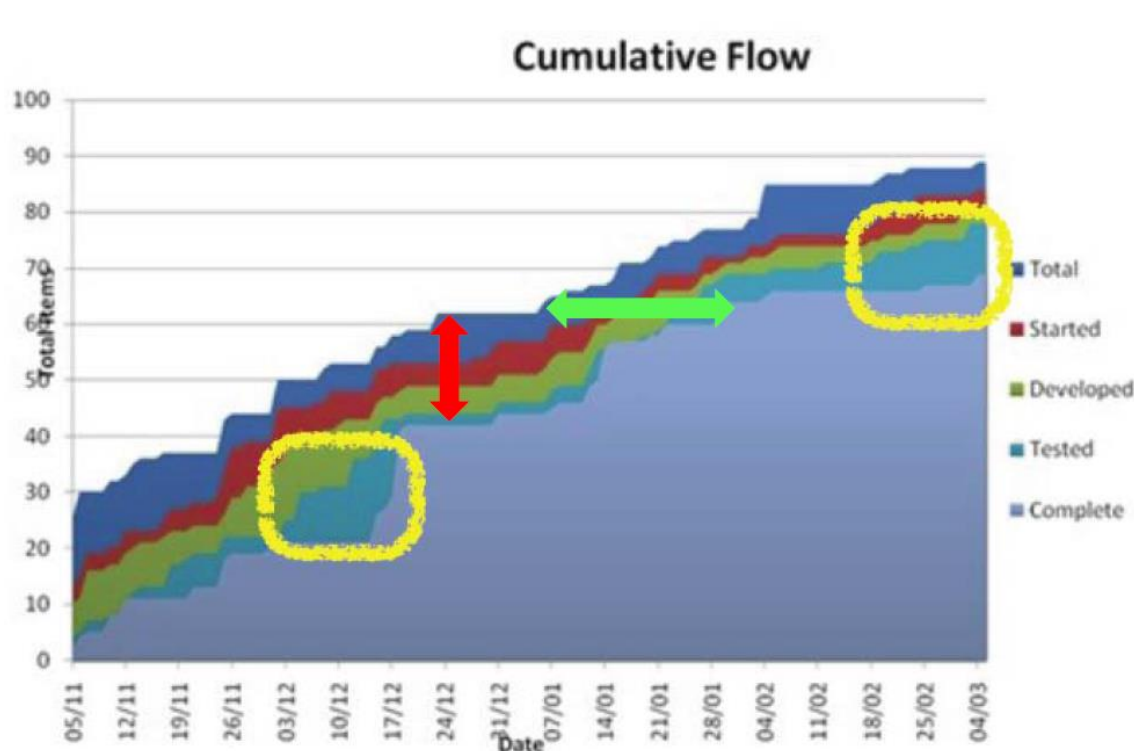


Figure 10 Kanban Cumulative Flow Chart

By looking at the vertical distance of a chart we can define how many items are currently in progress. The horizontal distance shows how long it takes for a task to be completed. Measuring the horizontal distance on a Cumulative Flow Diagram allows you to monitor

the Cycle Time, according to which you can make a prediction of when all the work in progress will be done. Vertical distance helps you to set the right work in progress limits.

Cumulative Flow Diagram should run smoothly. Large steps and flat horizontal lines indicate impediments to flow or lack of flow. Variations in the gap or bands stand for bottleneck situations, which usually occur due to irrelevant work in progress limits. This means that the number of tasks in each column should remain at the same level over the time. In addition, too many tasks in the queue mean either problems with finishing work on time or that the employee on the next stage cannot deal with work.

3.5 Hybrid Project Management Approach

Hybrid Project Management approach is a method in which the requirements and release planning are done in waterfall method, this methods combining the waterfall and agile method for the project execution, as explained in the picture below is referenced from the article in reference [2], which In the first sprint the project planning and proposal is done in a traditional way of analysis, design and documentation and the product backlog items are made, in the second spring the agile scrum method is followed in the team to critical path prototyping is made and presented to the team and the stakeholders and also tested and bugs are found, and the product backlog is groomed according to the finding in the sprint, and the new backlog along with the bugs are made sprint backlog for the next sprint in which the final prototyping is made, and demonstrated to all the teams involved in the organisation. Application development teams uses the approach similar to this hybrid project management method.

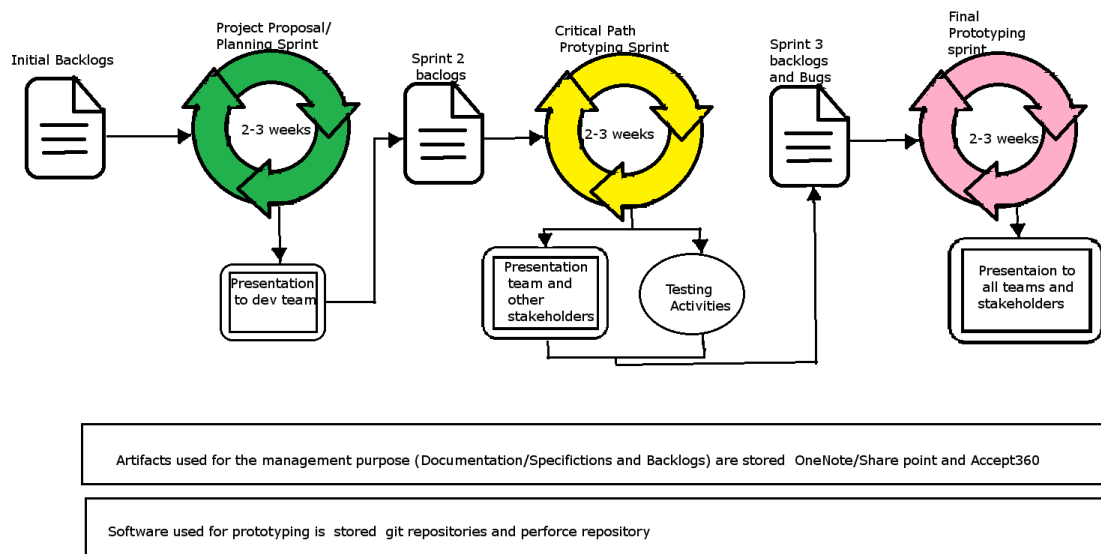


Figure 11 Hybrid Project Management Skeleton

3.6 Agile Maturity Model

Over the last decades the CMMI models has been used most predominantly to measure the quality and maturity level of the organization, ever since recent times when the agile software development came in to existence in most of the software development companies, where the customer needs are changing rapidly, there has been many studies done to analyse the adaptability of CMMI for the agile method of working,.

Main objectives Agile software development methodology are lower cost, high productivity and satisfied customer. The CMM tends not to focus the software process on an organization's business objectives in their software process improvement programme [7]. Also most companies, small to large companies found it is too difficult to reach higher levels in the CMM [7]. The study also mentioned that the CMM improvement path is not always smooth, the efforts generally took longer and cost more than expected. While agile software development methodology is targeted to lower cost. Some of the KPAs have been found difficult to apply in small projects [7]. This may be because CMM was originally structured for big enterprises [7]. CMM addresses practices such as document Policies and procedure that large organizations need because of their size and management structure [7]. Hence there is new innovation which were created to measure the levels of Agile software development practices which is called Agile software maturity model

The Figure 12 below from the reference [7], states the different levels in AMM, in simple terms from the article the levels could be defined as below Level 1: Initial Level is where the organisation unstructured and has no process improvement goal. Level 2: In this level the organisation is has project or software planning, customer or stakeholders orientation practices. Level 3: Defined Level is about having Customer satisfaction, Software quality and development practices, this level denotes a more focus on practices related to customer relationship management, frequent deliveries, pair programming, communication, coding, testing and quality of software. Level 4: Improved Level is to have People orientation and project management Practices, Companies at this maturity level are in a position to collect detailed measure of the software development process or practices and product quality, both the software development practices and products are quantitatively understood and controlled using detailed measurements examination of risk and respect to the team who is going to develop the system. The AMM at level 4 maturity aims to help developers or managers to respect for the co-workers or people involved in the project, identify and improve problems related to team sustainable pace and organising team by itself. This is achieved by an assessment of current process and to identify where weakness lie. Level 5: Mature level is when the organisation has Performance Management and Defect prevention practices in place, Companies at this level continually improve their processes through quantitative feedback from the process and form testing innovative ideas and technologies.

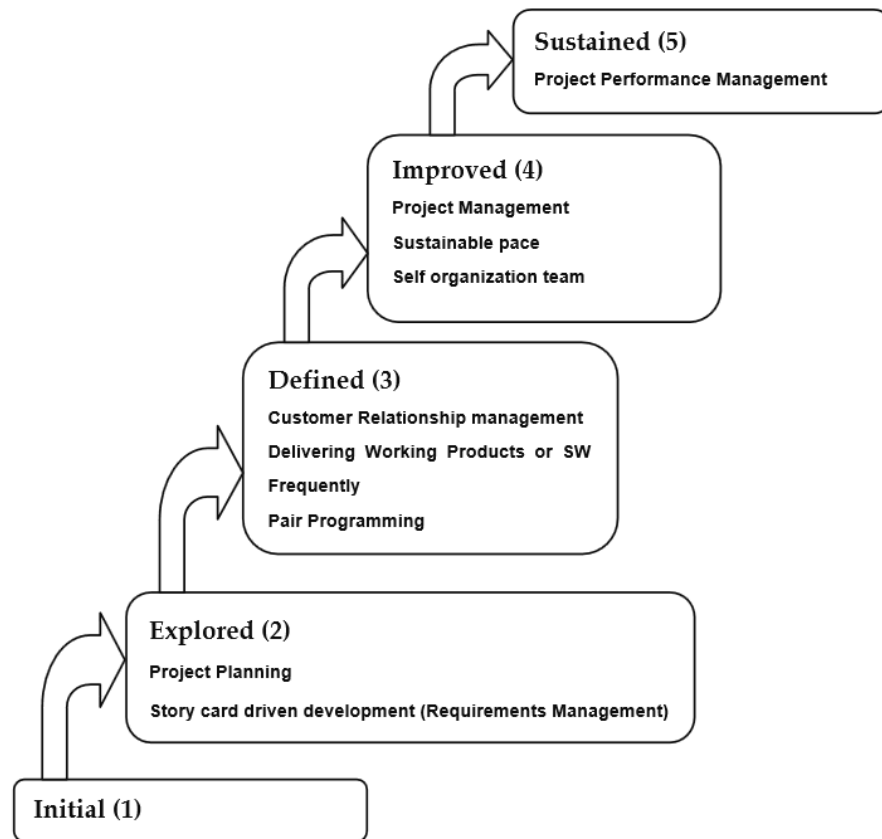


Figure 12 Agile Maturity Model Levels

3.7 Agile Software Product Management

The software product management (SPM) includes the process of managing the requirements, defining the release, defining the context for the products involving internal and external stakeholder, and, this topic also includes many other areas, many process are followed as this is more driven by market driven requirements engineering and currently there is very little of scrum done in the requirements engineering. The article is a case study on software product management, described in the reference [3], proposes the use of an agile SPM method based on SCRUM. It also prescribes the use of two different sprints, one for requirements engineering and the other one for development engineering.

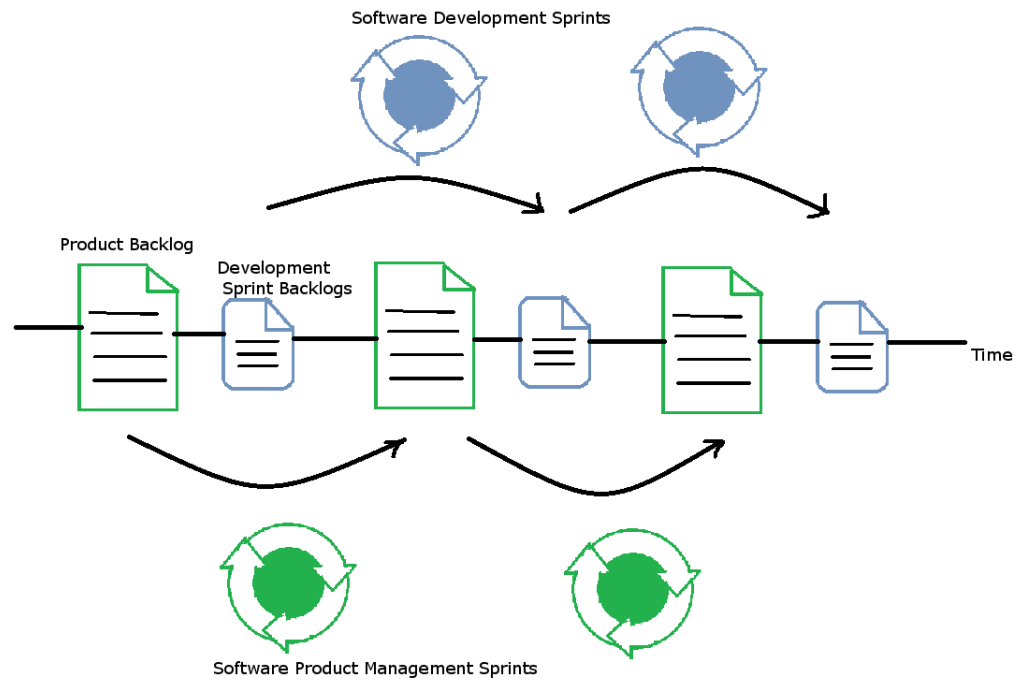


Figure 13 Agile Software Product Management

3.8 Accept360

Organisation which belonged to Nokia smart phone division, before the merger with Microsoft used the Accept360 tool as the key tool for requirements and release management. The tool has features to support requirement engineering and also the software development engineering. It is a web based tool which is supported in Internet explorer and Firefox browsers, in Desktop computers.

3.8.1 Accept360 Elements

Accept360 has defined elements for requirement and release management, they are called Roadmaps module, Requirements Module and Teams Module. Roadmaps Module is used to for managing the release milestones and artefacts related to that. Requirements module provides features to manage the contents of the releases. Team module provides support to plan and allocate the resources for working on the planned contents of the releases.

3.8.2 Accept360 Team Element Ranking Tab

Ranking tab contains user interfaces for managing team sprint backlogs.

It has the team element for the planning and managing the sprint backlogs. The ranking tab has two tabs the Backlog pane and Sprint Pane .The Backlog pane lists the backlog items for the project, which could be easily dragged to the Sprint Pane, which contains the list of selected backlog items for the selected sprint.

3.8.3 Accept360 Recommended Practices

For setting up the Accept360 for an application, it is recommended that the contents added for any application follows a certain recommended format, it is recommended that the backlogs are added in a standard hierarchy and the structure of the contents is recommended to be as in the Figure 14. Each application has a folder with the application name, under which sub folders are created for each version of the application and application driver if it exists.

Structure In practice

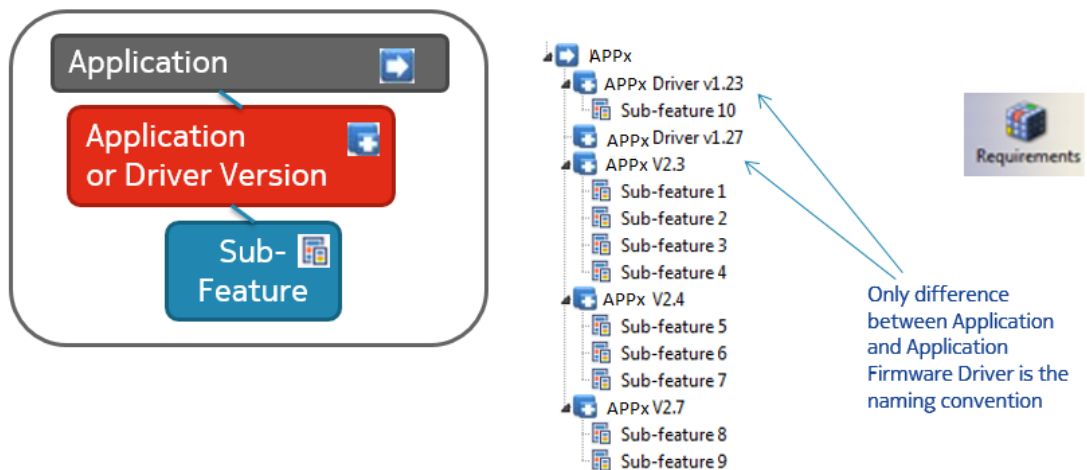


Figure 14 Accept360 Structure in Practice

3.8.4 Accept360 Responsible Actors

As explained in the Figure 15, at every stage of the project the contents needs to be updated in the Accept360 by the program manager or the project manager at some stage or by both of them to together. And it emphasis that he Project manager has the overall responsibility to ensure that the contents in the tools are up to date.

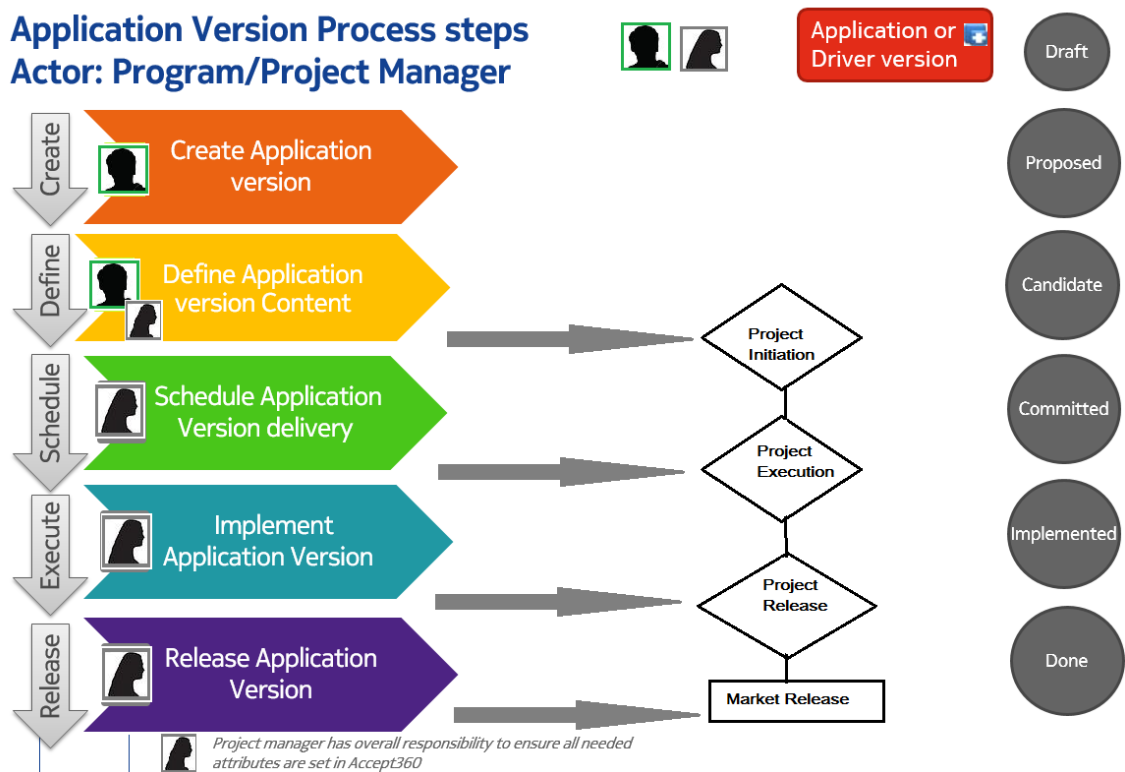


Figure 15 Accept360 Actors

3.8.5 Accept360 Agile Task board for Scrum features

Agile task board provides the features that are needed for supporting the task planning by the scrum teams. Accept360 tool has support for complete scrum process. Key components that are used in scrumming with Accept360 are Feature, Sub-features, Stories and Tasks. The Product owner creates the Feature, Sub-Feature and Story contents. The scrum master has access to Stories and Tasks. During the Sprint planning the stories are pulled for each sprint and the Tasks required to complete the stories are created

by the scrum master after discussing with the team members. And the Developers have access to update the status of the tasks.

Feature in the Accept360 can be compared to the Theme in scrum terms, but more equivalent in usage is an application as a whole is represented as a feature, and the Sub-Feature as Epic, which is a list of all the requirements for the application. When the sub-feature is drafted, it is then proposed for implementation, and stories are created, and then the stories are assigned to the teams, as sprint backlog items, which then gets a story point assigned for it. For each stories the tasks are created and is assigned to the team members. Below table represents the lifecycle states that are available in Accept360 and only some as states are used in defining the states of the requirements in the ASW projects.

SW Lifecycle	SW
Draft	Initial status when a new item is created first time; not yet ready for further actions and not yet in any backlog (with schedule)
Proposed	NOT USED
Candidate	NOT USED
Committed	Development team has committed to deliver item by Planned Delivery Date for certain programs.
Implemented	Item has been implemented and tested by the responsible team and given to the integration team.
Done	Item is ready i.e. it has been implemented, tested and integrated.
On hold	NOTUSED
Rejected	Item rejected

Table 1 Requirements LifeCycle

3.8.6 Accept360 Kanban features

Requirements can be synchronized from Accept360 Team Backlog to 'Backlog' column of the Kanban Board, sorted by Rank order. From there it is possible to prioritize items (ranking is updated to Accept360 UI accordingly, see who's responsible, Track changes, Balance workload and limits, also it is possible to configure team development states as needed, each column can be bind to Accept360 Lifecycle, progress can be also followed in Accept360, it is possible to set WIP limits for each state to avoid unfinished tasks and visualize the bottlenecks, emphasizes to have a work item 100% done instead of having many 80% done, is has to be noted that current Accept Kanban Board version supports only Requirements management, not Tasks.

3.8.7 Accept360 Nzilla Support

Accept360 supports importing Open bugs from Nzilla to the corresponding team. It helps the development teams to give relevant information from Nzilla about the bug to enable them to Rank the related Bug against other Stories on the Team Backlog, It is a One-way integration and Defect master always in Nzilla. Nzilla enables Lifecycle mapping and synchronize between tools to improve visibility. To integrate Nzilla with Accept360 a separate structure has to be created in Accept. In scope Programs and Component level automatically Synchronized using Nzilla Database IDs .Any name changes will be automatically updated, this stop the solution breaking and make deployment to teams easier. Only downside is there will be some extra Bug Bags and folders for teams that may not yet be using the solution.

3.9 San Diego Team Practices

The teams in San Diego had been able to prove that they have standard way of usage of the Accet360 tool, to visualize the needs and progress of the project, in every stage of the project. Here is the snapshot of the way the team has defined the sub-features, which includes a clear steps and functions that needs to be done in order to execute a

project, starting from the initial prototyping, until the planned feature or application is available in the market.

The items in the Requirements module have the prefix to each item which explain the common activity that is need to be executed in the project and also there are items that explain the main stages which a project would go through, so it helps to visualise the project situation with the items in the Accept360 tool.

Goals of every functionality are also clearly included in the list of sub-features and stories. The team follow the template available in Accept360 tool, below is the screenshot of the template.

Also the teams follow the mixed Scrum and Kanban methods. The teams have sprint planning, Daily scrum meeting and Retrospective meetings. And meetings are helped whenever it is appropriate and needed, and the team has a regular practice of keeping the Accept360 updated always.

3.10 Making Most of Scrum and Kanban

As the saying goes, No tool is complete and No tool is perfect, it only depends on how it is used, this applies to both scrum and Kanban. The value of a tool is that it limits the options. A process tool that lets user do anything is not very useful, this process could be named “Do Whatever”, but having a “Do The Right Thing” process is guaranteed to work.

Scrum and Kanban have both of their own pros and cons, so it is good to understand the difference between the two tools, so that we could make the best use of both the tools.

Difference between Scrum and Kanban

Scrum	Kanban
Time boxed iterations prescribed.	Time boxed iteration optional. Can have separate cadences for planning, release,

	and process improvement. Can be event driven instead of time boxed.
Team commits to a specific amount of work for this iteration.	Commitment optional.
Uses Velocity as default metric for planning and process improvement.	Uses Lead Time as default metric for planning and process improvement.
Cross functional teams prescribed.	Cross functional teams optional Specialist teams allowed.
Items must be broken down so they can be completed within 1 sprint.	No Particular item size is prescribed.
Burn down chart prescribed.	No Particular type of diagram is prescribed.
WIP limited indirectly(per sprint)	WIP limited directly (per workflow state)
Estimation prescribed.	Estimation optional
Cannot add items to ongoing iteration.	Can add new items whenever capacity is available.
A sprint backlog is owned by one specific team	A Kanban board may be shared by multiple teams or individuals.
Prescribes 3 roles (PO/SM/Team)	Doesn't Prescribe any roles
A scrum board is reset between each sprint	A Kanban board is persistent
Prescribes a prioritized product backlog	Prioritization is optional

Table 2 Difference between Scrum and Kanban

3.11 Scrumban

Scrumban is a result of the thinking to get most value of the scrum process, like Kanban. Scrumban is a pull-based system, where the team no longer plans out the work that is committed to during the planning meeting, and instead continually grooms the backlog. The same Scrum meetings can and should still take place, but the cadence of them can be more context-driven. The real key factors for Scrumban, though, is ensuring that work in progress (WIP) is still limited.

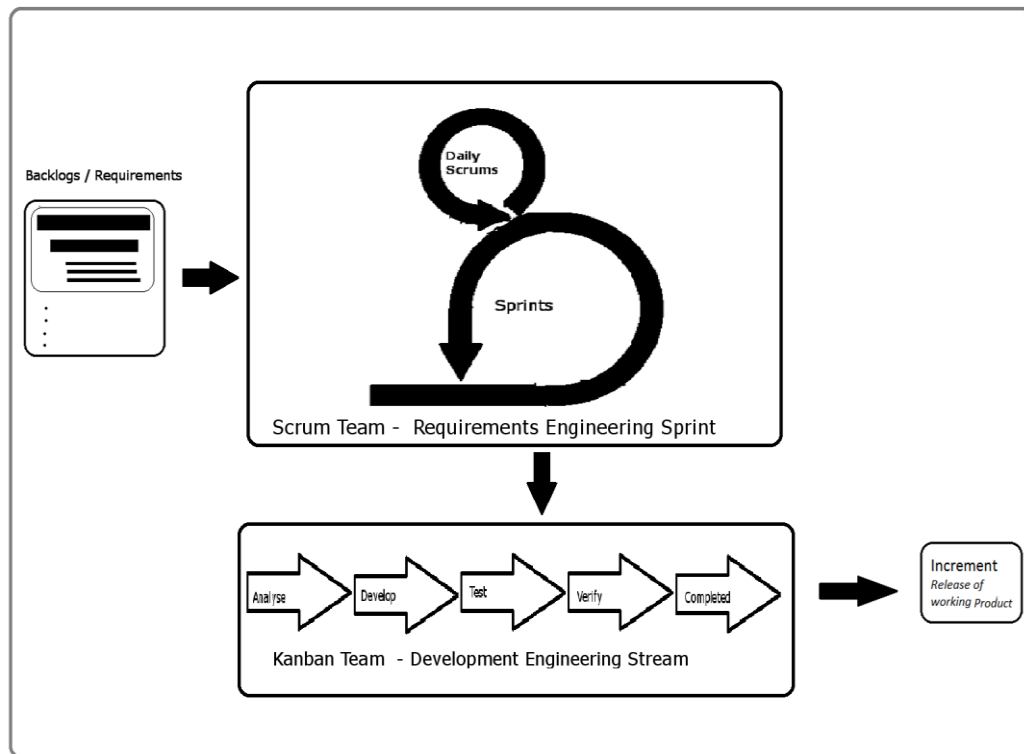


Figure 16 Scrumban Sprint and Value Stream

Scrumban works with the Work-in-progress limits, not Sprints. With Scrum, the amount of work that is ongoing is limited by the Sprint time commitment. But in Scrumban, with no specific time commitment, the team must limit itself through the use of WIP limits on columns within their task board. The goal is always to move tickets in a flow from left to right on the board. If too many issues are in progress, the team is at risk of not finishing anything to high quality standards. Instead, there should be a maximum number of tickets allowed per column. If the number of tickets in that column ever exceeds the maximum, the entire team should swarm onto that column and help move tickets on. This should happen no matter what functional role a team member fills.

Scrumban still supports the scrum ways of having the meetings and time boxed deadlines for the release of the features, planning meetings should take place as often as they are needed. When the team is unable to regularly pull stories off the top of the backlog at their normal pace, a planning meeting is necessary. Review meetings helps to improve the way of work with feedbacks. Reviewing work with clients and customers is the only way that development teams can get the feedback necessary to properly

adapt what they are working on. Clients tend to prefer that these are held at a regular cadence.

Retrospective meetings. These can vary when held, but a general rule of thumb is to hold a retrospective after every review. This is the most useful part of the Agile process and should be given the proper place for that.

Daily stand-up meetings in the Scrum world follow a simple pattern. The team takes 15 minutes and each person says, a) what he/she did yesterday, b) what he/she is working on today, and c) what is blocking any of that work. In practice, this boils down to redundant statuses that recount information available on the team's task board. For Scrumban, a more effective method is to refocus on the flow of tickets on the board. That same pattern of yesterday/today/blocked can be transferred to the tickets themselves—the group moves through each column and briefly discusses each ticket and what is necessary to move that ticket rightward on the board. This provides far more context to the team and informs every one of any major architectural or design decisions.

Metrics can certainly be useful, it interprets the complex process that is going on in the team, it is not just a single value or number to be expected to know how the team is progressing, and it could help visualize the situation the team is going through on the way to achieve the target. The term Velocity, the amount of story points a Scrum team completes in a single Sprint, is such a metric that incentivizes lower quality at the end of a Sprint as a team scrambles to finish every last story they committed to. When the number fluctuates, as is common with a newer team, the stakeholders begin to question the outputs of the team, and even the effectiveness of Agile itself. In Scrumban the metric is cycle time instead of velocity. This is the length of time a ticket takes to complete, measured from when it is first began. Over time, a statistical analysis of all tickets in the project can yield a mean cycle time and standard deviation. This can be a useful planning tool at a macro level, as it is trivial to add up the number of stories and multiply by mean cycle time.

3.12 Summary of Best Practices

From the study and discussion about the best practices, it is found that the scrum method is used mostly in the software development teams as it is very good for planning and scheduling meetings, but it also recommends that the sprint has to be pre planned and should not be interrupted in between and the backlogs need to be reset, which makes it to a situation that we cannot keep more buffer for backlogs to be tried and everything need to be reset at the end.

Next that Kanban is used most widely in software development organizations, which works with Work in progress limits are the visual indicator for the execution and planning of activities. And also recommends that that designing of the Kanban and steps for designing should be followed, which if not followed would result in a situations that he project situation could not be in control.

Next the thinking of making most out of the Scrum and Kanban has been received as a good approach to make use of good things in both the methods and which is best suitable for the needs in software development organizations, this could well be implemented with the techniques defined in the Scrumban.

Next the hybrid project management models is a good method which gives a way to have the traditional way of software development methods, and also have scrum included, but at that same time, this does not address the needs that the planning need to be dynamic, not only the execution is dynamic, so to have agile mode of working for planning the requirements and developing the requirements, that agile product management method is very suitable.

Next the study of the Accept360 tool details that, the tool has a lot of features in it, but it is built in a way that, the tool need to be adapted but adding more plugins to it ,for each feature, to be able to utilise it more efficiently .

From the study of usage of the Accept360 in the team in Sandiego, it is found that the content what is used to describe the project requirements or the backlogs play a key role in visualizing the project status, and also it make it clear that defining the contents of the

backlogs and work items in a standard way it is done in the team, would make it more helpful to know the needs and status of the project easily.

4 Current State Analysis

Current state analysis was done in the teams in Espoo , two teams were analysed and in each team the Product Manger, Project Manager , Development Team Lead and The Quality Lead were contacted and interviewed on the following topics: Project Details, Team Information ,Agile Practices,Project Planning and Execution Information,Project Situations,Suggetions for desired Improvments in Accept360 tool. The interview topics and questions are listed in the Appendix 1 .

4.1 Analysis of teams in Espoo

4.1.1 Analysis in Team X

Team X is working on creating a new version of an existing application, so this application is expected to replace the old application mainly with the new user interface and improved usability of features, the development phase of the application would need a complete cycle from proto typing the new user interface and adding all the functionalities to the new applications. This means that the efforts needed for developing the working application could be estimated to an appropriate value, which would help define a process to follow.

The team uses the scrum process for planning and execution of the activities. Accept360 tool is used during the sprint planning and review meetings. UI specification document and the story description in the Accet360 serves as the documentation needs of the projects, as they are updated with appropriate information. UI Design team shares the updates and synchronize their work for their sprints with the spirits of the development team sprints and meetings. Also a short description of key features of the application is documented in share point documents management page.

Team follows two weeks sprints and at the end of the sprint there is a demo and pre-planning session for the next sprint. Releases and Testing are done weekly during the project execution stage.

Team uses the Agile task board in Accept360 for managing the tasks, also the Nzilla is used to check the issues and bugs are also worked on as tasks. There are no daily meetings in the team, it is expected that each team member is located in the same location and has to contact the development lead or another team members to know what has to be done and what is expected.

The discussion with the key stake holders of the project, the Lead Program Manager, Project Manager, Lead Developer, and Quality lead is available in Appendix 2-4.

4.1.2 Analysis in Team Y

Team Y is working on developing value addition features for the existing application, the Team includes many small teams, every feature addition and implementation is initiated whenever the decision to have the feature in the application is accepted and is prototyped by one of the small teams and is integrated to the application.

Team uses the One Note as a tool for planning and updating the backlogs. The release plan, requirements list and status is tracked in OneNote pages. Accept360 is not used for these purposes, the reason explained was, the releases happened ever month and the features are released in very small time.

Team uses the UI specification as the document to know the details of the project. Accept360, could not be relied on for the details of the project, as it is not fully updated all the time.

Sprints are mostly weekly sprints, sprint meetings are held every week, and Adhoc meetings are organised whenever there is need. At the end of every week, an open agenda meeting is held, which is for having an open discussion about anything related to project and or give presentation about the progress. Testing actives are done on weekly basis.

The team had previously tried using the Kanban for their planning and scheduling before, but it was not very helpful, as they did not have control of the activates and work items all the time, as it ended up in just having a board with full of task cards and nothing going in planned way and things went just out of hand, but what it is important to note here is, the Kanban followed was not a designed to be adopted for the overlaying process in the organization.

The discussion with the key stake holders of the project, the Lead Program Manager, Project Manager, Lead Developer, and Quality lead is available in Appendix 5-8.

4.2 Summary of gaps between best practices and current practices

One of the main aim of the Agile methodology is to have clear visibility of the project at any point of time, so that any kind of resources are needed deliver that project as planned, are addressed at the earliest, this could be achieved only if the teams follow the prescribed practices of the standard methods they adapted to follow or follow a methods which would not prescribe more rituals, but still is would make the situation more visible to all the stake holders. But currently, the teams do follow standard methods, but skip some of the recommendation in the methods.

Though the teams are very dedicated to deliver the products and functionalities, the team members are not interested to follow the best practices and tend to work in more self-centred way, because of which certain practices which would help all the other team members to perform well are not taken in to consideration. The teams have their own way of working, than following the standard practices in the way of working,

From the study it is found that there are some practicalities that are not followed in the teams, a clear understanding of responsibilities and who is responsible for updating the tool is missing. This is also obvious from the contents in the items in the Accept360 tool, which is not up to date for some projects.

Also there are some needs that are missing in the current processes and tools, for example the Product backlogs have to be synchronized with the tasks to add the tasks for the sprints, there need to be a story to add task, but in practice the development teams

have more tasks, that need not necessarily have to be a requirement backlog item, but is needed for their day to day activity. The requirements engineering and the software development engineering are in this way tied more closely in the tool. Even though there need to be synchronization between them, the tasks in the development engineering team need to be created according to the development activities, to get the requirements implemented, this would need different terminologies in defining the tasks, that would not fall into different the category of work items and tasks definitions

The Agile task board is updated only at the end of the sprint, which explains the board is not actively used in the team, the reason for this found from study is that the way of working is not the same in the flow of states in the task board, also there are tasks which could go longer than the sprint time , but in scrum after every sprint they had to be reset and added to next sprint or put as impeded, which would give a picture that there the things are not going well, but in general there are some dependencies that the tasks need to stay between the sprints, for it be completed in a better way, it is because the order in which the task need to be worked on changes, that some tasks need to be put on hold for some time waiting for other tasks to be complete or started. As the sprint is time boxed and needs a reset of the tasks and backlogs for the sprint after every sprints, it leads to a situation that the tasks are updated in a way that their status is not related to actual status and is updated for the purpose of resetting the sprint backlogs which scrum recommends. So there is a need for other methods and tools like Kanban which would help the developers to keep the tasks across sprints without resetting it at the end of each sprints. But it has also been experimented in the Team Y, that Kanban method alone was not helping enough as there is a need to time box the release of features at some point of time and the releases are tied to the release of the devices to the market, this needs some kind of iterative approach as well, for planning the activities, hence there is a need for using some of the features from the scrum method and Kanban tool to get the needs of the project execution addressed completely.

5 Proposal

5.1 Theme Proposals

As summarized in the best practices and current state analysis, the key areas where the gap exists in knowing the status of the project is, how the contents of the project planning

and execution tool are defined and used, the difference in work items needed for requirements engineering and development engineering, and difficulties in combining the time boxed sprint development method with the continuously improvement and integrated development mode of working. These gaps could be address better by the themes that are explained in the below subheadings ,which would help fill the gaps that exists and help improving the visibility of the project status to all stakeholders of the project.

5.1.1 Usage of Structured Contents for Work Items

Work Items as explained in the best practices are the items like epics, stories, tasks, mmf and activities. The description of these items, could serve as documentation for the project, and this could include details about goals, features of the product being developed. All the work items would fall in one of the categories of the activities that are commonly needed to create the end product, like Functional requirements, Non Functional requirements, User interface design and development, Core technology implementation and interfacing, Performance validation and improvements, Testing and Error correction. The organisation in case company in which the research is being done, has a recommendation and guide for creating and maintaining the work items. The guide provides information about required structure for defining the work items, naming and versioning conventions, avoiding duplicates in contents, this guide could be used as reference to maintain the contents in a standard way. The guide also includes details about environment needs to use the tool, such as browser requirements for the tool, plugins needed to have a better performance of the tool, and it includes a template which explains the backlog creation ethics.

Initiatives to use the guidelines and keeping the work items updated, would help improve the visibility of the project status, so that by looking at the content of the sub-feature, stories and tasks it is possible to have a clear picture of the situation of the project. For example the Team S in Sandiego have the contents defined in a specific way, they follow different syntax for defining the work items, so that when looking at the contents in the Sub-features, Stories and Tasks, it is possible to understand the stage of the project, So keeping the contents more meaningful in the recommended standard way, could make the visibility of the project situation more clear to all the stake holders involved.

For example the Figure 17, show the sample content from the best practices studied in the San Diego team, this could be used as reference for creating the items for the scrum team backlogs. The template has suffixes such as Story , Demo story, UI design, Miscellaneous UX,UI Spike, Automated Testing, Exploratory ,Reliability ,Compatibility ,Interoperability ,Non Functional ,Certification Testing and other stages the project need to carry on for finally delivering the project. These suffixes would give more information about what is planned and being done at that stage of the project, so this helps to have more details of the activities carried on in the project and it would help to provide needed resources for those activities based on their needs.

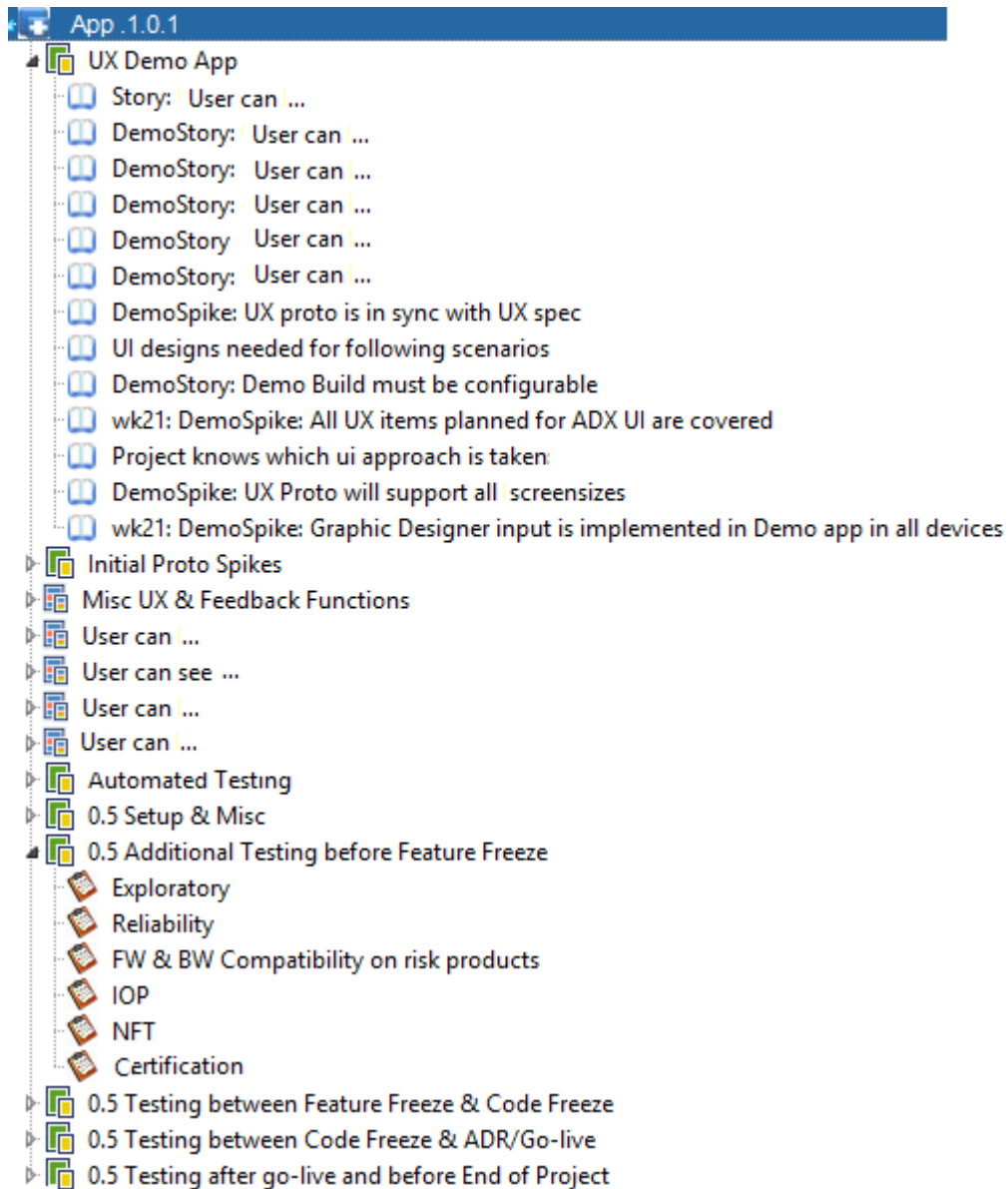


Figure 17 Template for Work Items in Scrum team

5.1.2 Usage of Agile Software Product Management

The software product management (SPM) as explained in the reference [3], proposes the use of scrum in product management. It also prescribes the use of two different sprints, one for requirements engineering and the other one for development engineering.

For a scrum process the backlogs are the key instruments, and there are two different kind of backlog items involved in software development, the Product backlog, which is the prioritized list of requirements, that includes all relevant needs of the product that is being planned to be developed. Some of these Product backlog items are then copied by the development team as their sprint backlogs called Development Sprint backlogs, which are then split in to stories and tasks.

Though there is a tight dependency between the two backlog items, the backlog items picked up by the development teams are reset every sprint and different tasks are created newly. Hence there is a good need that these two backlogs items are used differently by the software management teams and the software development teams. Currently scrum is followed only by the development teams, so as explained in the reference [3], the article suggests the use of agile scrum methods in defining the Product backlogs would be more efficient to manage the change and synchronize the changes and progress, in both requirements engineering and development engineering.

From current state analysis interviews in the case company, it shows that the existence of the difference in requirement engineering and development engineering needs for managing the backlogs, even though there has to be synchronized usage backlogs between these two teams, their functional needs demand the need for having two different sprints or cycles for the same product.

5.1.3 Usage of Scrumban

The requirements engineering teams which involve the project management and planning requires clearly what is prescribed in scrum, the sprints, roles and ceremonies, but at the same time the development engineering teams need a stream which is more like Kanban to keep their flow of work steady and visible, so Scrumban which recommends work pull system, which is a scheduling method based on demand and at the same time recommends to have the scrum way of planning and checking the situation by the aid of meetings, would best suit the needs in the organisation in study.

There are also many teams involved in one project and each team have their own planning and execution cadence, the Figure 18 is a sample framework, which explains the involvement of different teams, with different work cadence defined, for the same project and more relevant for scrumban method. The frame work is also based on the concept explained in the reference [9], which is good for complex implementation. Currently there is only one cadence which is defined for the development team and other teams like requirements management and UI design teams ,which are very tightly working with their deliverables as input to the development engineering team, do not have a designed cadence for delivering their deliverables and they are provided in a dynamic and not planned way, so defining a cadence and process method for all the key teams, that are involved in the project, would make the interaction and situations much clearer and easier for planning and executing the activities of the development team and for the whole project as well.

As an example there are three teams in the sample framework, first team is the team which is the key team that creates and defines the requirements of the features of the product being developed, the product needs several other teams also to be involved very closely to get the product developed fully, the development engineering team and the UI design teams are the other key teams which work on the creating the product, as in the Figure 18 the Requirements team follow the scrum methods ,they define and provide the backlogs for the other teams, and also schedule the planning and status meetings, which could for example happen in a three weeks cadence, and the development team have a cadence of two weeks for releasing the working software product, and the UI design team works in flexible cadence of delivering the UI design document whenever there is a need for new method of defining the user interface, all these three teams has to work independently, but still depend on very closely on each other, so having a framework designing like in the sample framework in the Figure 18 would help planning and coordinating the teams in better way, also this would the help visualize the dependencies on deliverables between teams and define them in a structured ways and this would help all the stakeholders involved in the project to visualise the status of the projects more clearly.

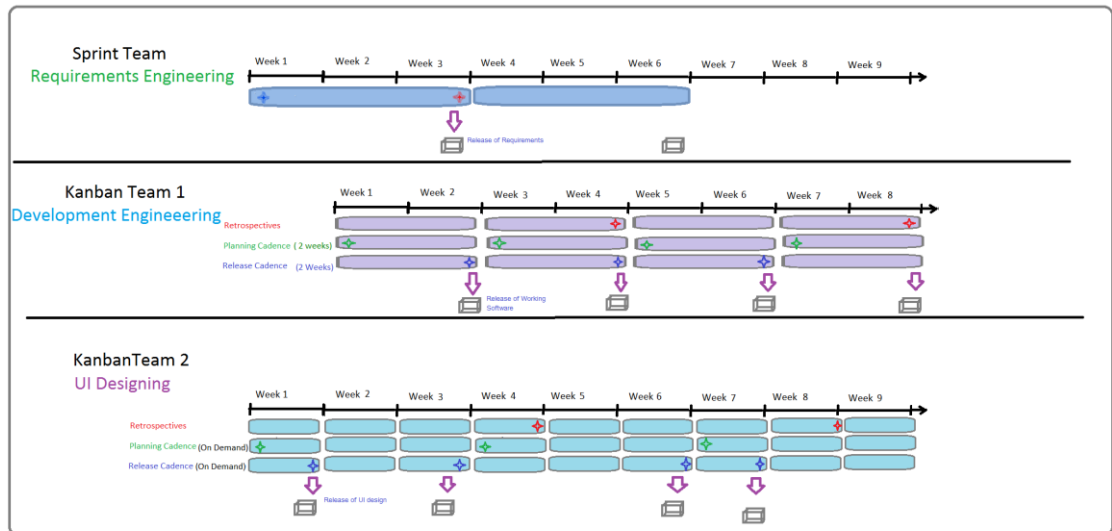


Figure 18 Sample Framework with Scrumban Method

In Scrumban the story and tasks card need to be built and maintained appropriately otherwise it would not yield any value, as in the Figure 19 below, the story cards or the sub-features in green are the items which are worked on by the requirement management team, which mostly involve the project managers and the development team lead, the stories are selected at the beginning of each sprint and story board is reset at the beginning of each sprint.

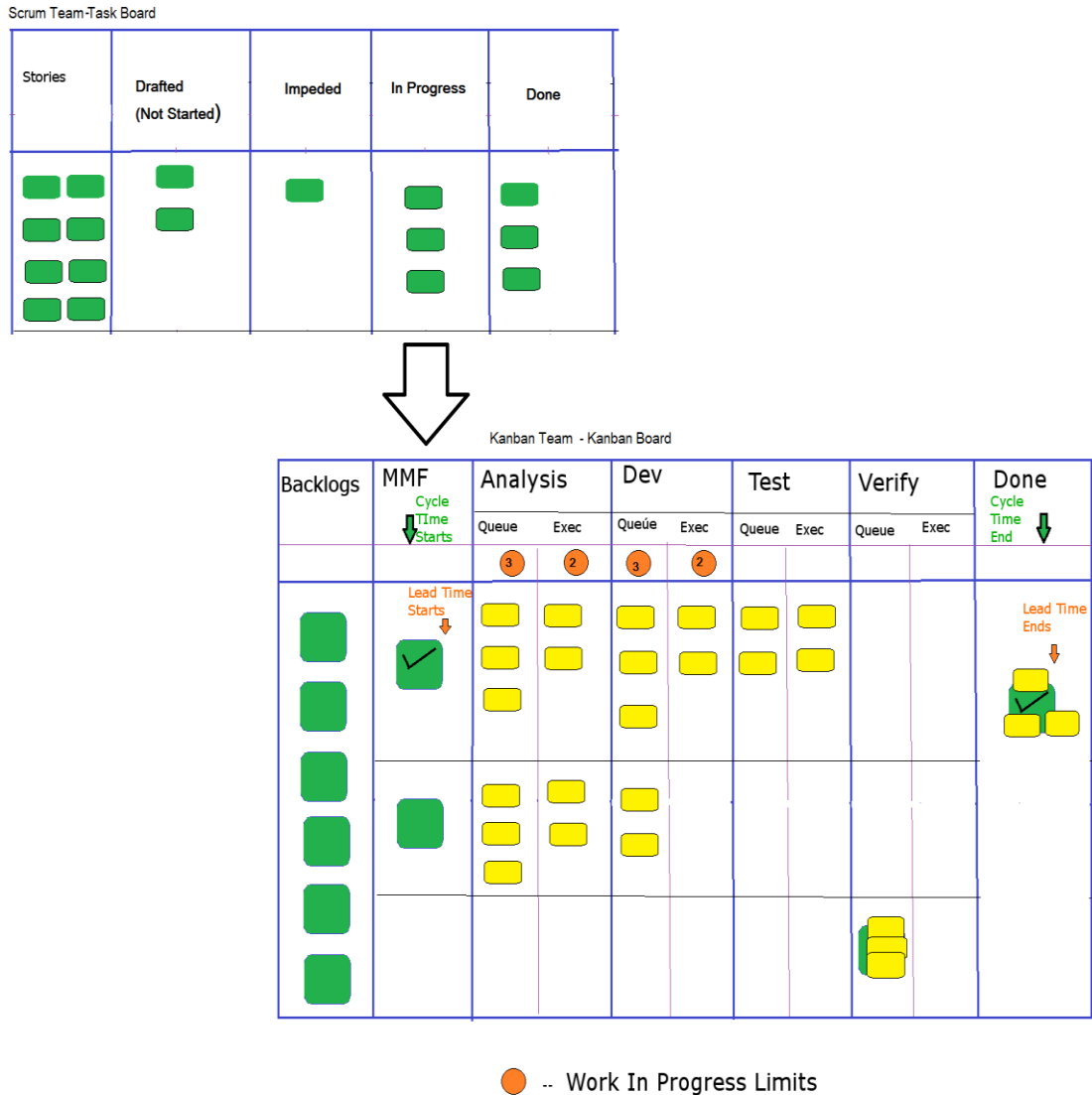


Figure 19 Sample Scrumban Board

Scrumban for the development team works with the Work-in-progress limits, not Sprints. With Scrum, the amount of work that is ongoing is limited by the Sprint time commitment. But in Scrumban, with no specific time commitment, the team must limit itself through the use of WIP limits on columns within their task board. The goal is always to move tickets in a flow from left to right on the board. If too many issues are in progress, the team is at risk of not finishing anything to high quality standards. Instead, there should be a maximum number of tickets allowed per column. If the number of tickets in that column ever exceeds the maximum, the entire team should swarm onto that column and help move tickets on. This should happen no matter what functional role a team member fills.

For the Software product management, Scrumban still supports the scrum ways of having the meetings and time boxed deadlines for the release of the features, planning meetings should take place as often as they are needed. When the team is unable to regularly pull stories off the top of the backlog at their normal pace, a planning meeting is necessary. Review meetings helps to improve the way of work with feedbacks. Reviewing work with clients and customers is the only way that development teams can get the feedback necessary to properly adapt what they are working on. Retrospectives meetings could be held when needed, or in regular intervals, so that team has an opportunity to give feedback about whole process involved and give input about what changes they want to have to, in order to improve the Efficiency of the team.

Scrumban can restore working time to the development team, and avoids unnecessary meetings. And most importantly, it can limit the team's work in progress, so that they can finish what they start to a high standard. Scrumban can remove overhead stress for the development team, increase efficiency, and increase the overall satisfaction.

5.2 Summary of Proposal

From the study it is understood that the usage of Accept360 tool for day to day activities of the project execution is not preferred ,due to the fact that the tools is very slow for accessing and updating the content. The tool is more suitable for planning the releases and maintaining features for devices globally, but it does not address the need for the software development projects that has quick and dynamic life cycle. But as long as the tools is used and since the tool supports many features of the agile development methods, the guides and training documents available for the tool are created to address the process followed in the organisation in case study over years, following the guidelines in the same way it is recommended, would help the teams to use the tool more efficiently for planning and execution, also would help all the stakeholders involved in the project to have the clear visibility of the project situation at any point of time.

From the current state analysis, it shows that there is interests to use other methods and tools in addition to methods and features supported in the Accept360 tool, and Accept360 is considered to be helpful more for requirement management teams, rather than for software development teams, so it is good if the teams could make use of other

tools suggested by the stake holders interviewed for current state analysis, tools like scrum works pro which already has synchronization with the Accept360, or the teams could try the tools like Jira and Version One for their activities and synchronize their sprint backlogs with the Product backlogs in the Accept360, but it is important to mention here that, for synchronizing backlogs with other tools the efforts and resources needed would be more. Hence it is good to further research on what tool could serve more good for the organisation's need.

Since in the organisation in case study , there is more clear work type splitting of requirements engineering and software development engineering, the Agile methods in requirements engineering would provide more visibility in project situations and requirements creation and grooming would be carried on in more structured and dynamic way.

Also from reference [8] and other online research papers recommend the usage of both Scrum and Kanban together for having more visibility of the project situation and progress. It would increase the productivity and efficiency of the team.

6 Conclusions

6.1 Summary

This chapter gives the general overview of the thesis, it is a look through of all the steps done in the research process. Validate the research by comparing the objective with the results, how applicable and generalized are the results in another context. And also check how reliable the research process is, if the methods and materials used in data collection and analysis are appropriate and whether this research could be carried on further by another person, in the same way it is done currently.

6.2 Practical Implications

The thesis was aimed to study the gaps that exist in current practices in software development methods and tools, in the organisation in case company, in comparison with the best practices in agile software development methods and tools, that are available from different sources and summarised in the literature part of the thesis, and to propose

the solutions would fill the existing gaps and would provide the ways to improve the visibility of the situations. The current state analysis was done by discussions with many individuals in different kind of roles, in three different teams. These discussion turned out to be a great source of knowledge and insight into the organisational process, mainly it gave an insight and good understanding of project planning and execution methods. Everyone involved were in highly responsible roles, hence the discussions were more clearly depicting the actual responsibilities and needs in a project, which helped to frame the proposals that would address the needs to fulfil their responsibilities in an improved way and help all the key stakeholders in a project to have better visibility of the project status.

The research was mainly based on the current ways of working, current issues and needs that were desired by the team members. The proposals are improvements that are needed in three different area of project planning and execution methods and tools, and proposals are framed based on the research articles in the references. The feedback about the proposal was very positive, that people involved in the requirement management showed interests in the proposals, and they wanted to have similar practices proposed in this research.

The current research focus only in finding the gaps that exists in the methods and tools and proposes the key ways that could be used to improve the visibility of the project status, and the research proposes the sample approaches that could be tried, to have better visibility of the project situations, which also help in reducing the delays in the project deliverables. The proposed sample approaches need to be designed further and piloted according to the needs in the teams, in the organisation under case study.

6.3 Evaluation

6.3.1 Objective Vs Outcome

The main objective of the thesis was to find the ways that would improve the visibility of the project situation to all the stakeholders involved in the project. In the study three main proposals were created to improve the visibility of the situation. First proposal is to improve the contents in the requirements management tool, so that the contents have the description of what is being done and what is the goal and outcome of the work item for

each requirement. Second is to have a separate scrum process for requirement engineering, which is called Agile Scrum Product Management, which lets the backlogs to be engineered with a process rather than continuously, so that they are created with better clarity, rather than in an Adhoc way, which definitely would give clearer picture and estimate of the project situations. Third proposal is using the Scrumban, which is having a scrum process for main process of requirements engineering and the other engineering teams like development engineering teams would work in accordance with their requirements from scrum team, with a key control agent as Work in progress limits. The development engineering teams would have a Kanban board designed, in a way that their work flow is checked by work in progress limits and is not reset every time the scrum sprint ends, in this way the actual situation of their work progress, could be seen in the work item flow and is more realistic. The outcome of all the above three proposals would create a situation where the process and the all the stake holders in the project are more closely involved, which would eventually improve the visibility of the project situations.

6.3.2 Reliability

Reliability checking for the thesis is done mainly based on the validating how well the methods and the materials used and well the outcome addresses the objective of the thesis. The Methods and materials used in this thesis, is mainly from the referenced articles and books, and the teams and people involved are from the organisation in which the case study is done. The objective and the outcome of the study are closely related to the real situation and more relevant to the study topic. In the literature part contents are designed in a way that the agile methods and practices are highlighted and explained in detail, that this report would serve the purpose of referencing as a guideline to design and improve them. Also current state analysis were conducted in a more generalised way, with more generic question about project planning and execution, so that the inputs from those interviews would give a generic overview of the process and the needs for improvements in the project, in this way this study has initiated a research which would further be continued in an elaborate mode, for example by clearly designing the scrum team in accordance with the recommended scrum ceremonies and designing the Kanban as per the needs of the project, and apply them in scrumban method and validate

the same with experiments in more teams, piloting the proposals, are some of the key research areas for future studies.

6.3.3 Validity

The thesis is validated using following methods, the Triangulation, Prolonged engagement in the field and Member Checking, further in this section. Also the answer to the research question “How the visibility of the project situation could be kept updated at any point of time?” “What are the best ways to stay on track of the current situation of the projects and so be able to manage the issues appearing earlier and minimize the delivery slippages?” Is answered in the proposal section and is addressed completely. Hence this states that the research has a consistent and logical approach in providing what is promised in as objective.

Triangulation is a validity procedure where the researcher search for convergence among multiple and different sources of information to form the themes and categories in the study. As a validity procedure in this method, the researcher uses a systematic process of sorting the data to find themes, by eliminating the overlapping areas. In this methods the researcher rely on multiple evidences rather than single incidents or data point in the study. The data collected include three different teams and ten individual discussion, also the best practices were studied from several articles, company guides and books.

Next method that is used for validating the study is, Prolonged Engagement in the field, since the thesis was carried out in the same organization where the researcher is working since many years and one of the teams which were involved in the study is where the researcher worked at the time of study for more than seven months , as Fetterman (1989) contends that "working with people day in and day out for long periods of time is what gives ethnographic research its validity and vitality. (p46)." the study has more vitality.

Member Checking method of validation, involves getting feedbacks from the participants, The feedback from the participants who were key stake holders in the organization in the case study and the Supervisor of the thesis who is in the role of the Operations Manager of the organization under case study were collected. The participant’s feedback states that the study was very useful in giving details about the agile methodologies and best

practices, in more elaborative way. The feedback from the supervisor was that the discussion and contents in the thesis were useful in understanding the current practices and also the proposals in the thesis are closer to the changes that were expected by many key stack holders in the organization, and was helpful to align and improve the current practices, for addressing current key issues.

Hence in the thesis, the qualitative method used for data collection, has provided with more valid information of the current state, the data from the discussions are valid and used to build the proposal. The research does not contain any personal opinions of the researcher, but is fully based on the data collected from the key stake holders in the organisation and the articles and book in the references. The researcher has used the experience in the organisation to be selective in collecting the data, so that the data is more relevant for the research, which has brought very good result in proposing the solution framework.

REFERENCES

ISBN-10: 073561993X

[1] NATHAN-REGIS B, BALAJI V. EVALUATION OF THE MOST USED AGILE METHODS (XP, LEAN, SCRUM). *International Journal of Engineering Science & Technology* [serial on the Internet]. (2012, Jan), [cited December 15, 2013]; 4(1): 23-29. Available from: Academic Search Elite.

[2] Baird A, Riggins F. Planning and Sprinting: Use of a Hybrid Project Management Methodology within a CIS Capstone Course. *Journal of Information Systems Education* [serial on the Internet]. (2012, Fall2012), [cited December 17, 2013]; 23(3): 243-257.

[3] Kevin Vlaanderen, Slinger Jansen, Sjaak Brinkkemper, Erik Jaspers, The agile requirements refinery: Applying SCRUM principles to software product management, *Information and Software Technology*, Volume 53, Issue 1, January 2011, Pages 58-70, ISSN 0950-5849.

- [4] Ken Schwaber. Agile Project Management with Scrum: Microsoft Press © 2004
- [5] Patel, Chetankumar, and Muthu Ramachandran. "Agile maturity model (AMM): A Software Process Improvement framework for agile software development practices." *Int. J. of Software Engineering, IJSE* 2.1 (2009): 3-28.
- [6] Sjøberg D, Johnsen A, Solberg J. Quantifying the Effect of Using Kanban versus Scrum: A Case Study. *IEEE Software* [serial on the Internet]. (2012, Sep), [cited December 17, 2013]; 29(5): 47-53.
- [7] Fogelström N, Gorschek T, Svahnberg M, Olsson P. The impact of agile principles on market-driven software product development. *Journal Of Software Maintenance & Evolution: Research & Practice* [serial on the Internet]. (2010, Jan), [cited December 15, 2013]; 22(1): 53-80.
- [8] Gross, JM, & McInnis, KR 2003, *Kanban Made Simple: Demystifying and Applying Toyota's Legendary Manufacturing Process*, Amacom, New York, NY, USA. [11 September 2014].
- [9] Effectiveness of kanban approaches in systems engineering within rapid response environments Richard Turnera*, Dan Ingolddb, Jo Ann Laneb, Ray Madachyc, David Andersond aStevens Institute, Hoboken, NJ, 07030, USA bUniversity of Southern California, Los Angeles, CA, 90089, USA cNaval Postgraduate
- [10] Henrik Kniberg and Mattias Skarin. Kanban and Scrum - making the most of both : Dec 21, 2009, InfoQ.com
- [11] Ladas, C. 2008 *Scrumban: Essays On Kanban Systems For Lean Software Development*, Seattle: Modus Cooperandi Press
- [12] Creswell, J. W. & Miller, D. L. (2000) 'Determining validity in qualitative inquiry', *Theory into Practice*, Vol. 39, Issue 3, pp. 124-130.

APPENDICES

Appendix 1. Interview Topics and Questions

Project Details	<ul style="list-style-type: none"> What kind of work is done in the team? <i>e.g Development of Applicatioans ? Services ? Quality enhancements?</i> 	•
Team Info	<ul style="list-style-type: none"> What is the Team Size? What is the Team Structure ? 	•
Agile Practices	<ul style="list-style-type: none"> What are the Agile practice followed in the team? scrum or kanban ? How often the requirements and the deadlines/milestones are planned/ updated? - <i>weekly planning? sprint planning ?</i> How are daily tasks planned /tracked/ updated? - <i>Daily Scrum ?</i> 	•
Project Planning and Execution Info	<ul style="list-style-type: none"> Team uses using accept360 tool ? How frequently the accept360 tool is used in the team? What are the other tools used in the team for tracking and execution of tasks? How do you use these tools for daily planning and updating the tasks? 	•
Project Situations	<ul style="list-style-type: none"> How is the team velocity calculated and followed up? What kind of deadlines are set in your team ? <i>Milestones / Releases/ feature completion?</i> How are the deadlines achieved in the team? - <i>All ways successful or Some delay? Earlier failures or delays to meet deadlines? Please give more details on this.</i> 	•
Suggestions	<ul style="list-style-type: none"> Do you have any suggestion to improve the tracing and execution of the projects in the current organization ? What improvements would you like to have in the accept360 tool ? Do you wish to have any improvemetns in the current way of usage of planning and execution tools to increase the productivity of the team? <p><i>Please give all your free thoughts that you feel might help improving the methods of project execution and that could help me define and create a proposal for the improvements in the tools</i></p>	•

Appendix 2. Interview

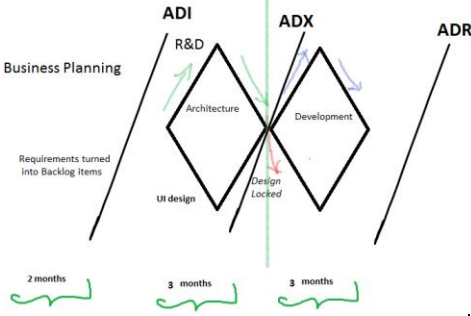
Date :-	16 th October 2013,	
Role :-	Senior Project Manger –Team S in Sandiego	
Project Details	<ul style="list-style-type: none"> What work the team is involved in? 	<ul style="list-style-type: none"> Development of applications that provide wide variety of services and features to customers.
Team Info	<ul style="list-style-type: none"> What is the Team Size? What is the Team Structure ? 	<ul style="list-style-type: none"> 2 Project Manager 14 Developers 2 Testers Product Manager ->Project manager ->Scrum Master/Team Leader→ Developers
Agile Practices	<ul style="list-style-type: none"> What are the Agile practice followed in the team? 	<ul style="list-style-type: none"> Team uses mixed Scrum and Kanban practices. Daily Scrums.

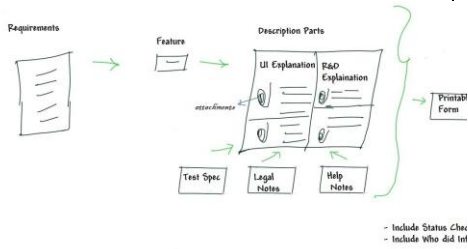
	<ul style="list-style-type: none"> • How are the requirements planned/ updated? • How are daily tasks planned /tracked/ updated? 	<ul style="list-style-type: none"> ○ Weekly Demo and Review Planning. ○ Scrum Retrospective sessions, to listen to the team and adapt accordingly. • Application Concept is finalised -> 1 Week Spikes Task done for ADI(Application Development Initiation) ->ADX(Application Development Execution) Development and Testing for planned time ,includes Rough UI Development , API Proto Typing with Simple UI test apps ,Integrating UI and API functionalities-> ADR (Application Development Release)
Project Planning and Execution Info	<ul style="list-style-type: none"> • Accept 360 Usage. • Could you explain what elements are used in Accept360 for planning and how it is used? 	<ul style="list-style-type: none"> • Accept360 is used by the team regularly. • Requirements Element is used to add the Stories(Sub-Features). Different types of Stories are Spike Story,UI Proto Stories,API prototypes, Feature/Funtionality ,Test Stories,Bug Stories. • Bug Stories are created for the bugs in the nzilla, they are manually picked up ,assigned and updated in nzilla. • Stories are added to the tools in such a way it explains what kind of items they are ,for example, each story has a prefix: , that makes the stories more understandable and trackable. • The prefixes are designed randomly ,e.g. SpikeStory: , UIProTypeStory: ,PREFStory: etc. • The SubFeatures are assigned to teams and can be prioritised un the Accet360 Tool.
Project Situations	<ul style="list-style-type: none"> • How is the Team Velocity calculated. 	<ul style="list-style-type: none"> • In the Pre-Planning stage always a buffer of 2 to 4 weeks is kept as buffer time for releasing the applications,this is done based on the approximation of the teams performance.This time is used for testing the applcaitons and as stabilization work for the application, in this way always the project are able to meet the release deadlines. • Getting a Burndown chart from accept 360 is possible, but still it is not easily done, as there are some delays in updating the status to the tool, everyone do not keep the data in the tool updated.Hence, manually the Project managers creates the burndown chart and asseses the velocity of the team and adapt accordingly.

Appendix 3. Interview

Date :-	22 January 2014	
Role :-	Lead Program Manager -Team X	
Project Details	<ul style="list-style-type: none"> What kind of work is done in the team? <i>e.g Development of Applicatioans ? Services ? Quality enhancements?</i> 	<ul style="list-style-type: none"> Application development.,as pProgram manager's the key re responsibility is defining the features and requirements.
Team Info	<ul style="list-style-type: none"> What is the Team Size? What is the Team Structure ? How many Teams ? If teams are located at different location ,where are the Teams located? How do they share work? 	<ul style="list-style-type: none"> Works with different teams in project X. Teams are located in Espoo and Lund. Program Manger works with different teams in project X ,three teams and some times even more teams work together,and coordinated the activities. Program manager's responsibility is defining the features and requirements.
Agile Practices	<ul style="list-style-type: none"> What are the Agile practice followed in the team? Scrum or kanban ? How often the requirements and the deadlines/milestones are planned/ updated? — <i>weekly planning? Sprint planning ?</i> How are daily tasks planned /tracked/ updated? — <i>Daily Scrum ?</i> 	<ul style="list-style-type: none"> Scrum is followed in the team. Team follow light weight scrum A practice of having handshake between scrums is followed,a short time period as of break s in between scrums is kept for this purpose of handshake activities. Sprint plan document is prepared and ready before the sprint planning. Backlogs are grouped for sprint planning. Backlogs are grouped into POTENTIAL backlog items, ACTUAL backlog items and FILTERED Backlog items , by the end of the sprint planning meeting.The picrure below explains more about categorising backlogs. <div data-bbox="1018 1415 1388 1736" data-label="Diagram"> </div> <ul style="list-style-type: none"> Discussions with the team on the tasks is done in sprint planning meeting and the team agree with the Format for the tasks and Rules for the tasks.

Project Planning and Execution Info	<ul style="list-style-type: none"> • How are the requirements planned ? • How often the requirements are updated or prioritised ? • Team uses using accept360 tool ? • How frequently the accept360 tool is used in the team? • What are the other tools used in the team for tracking and execution of tasks? • How do you use these tools for daily planning and updating the tasks? 	<ul style="list-style-type: none"> • Scrum strategy below is followed <ul style="list-style-type: none"> ○ <i>Monthly Vision for future</i> ○ <i>Brain Storming ideas</i> ○ <i>Consumer Insight (Understanding the customer needs and interest, for customers from different parts of the world ,explore by travelling)</i> ○ <i>Bench marking features and applications with other products.</i> • Excel sheet with set of formulas in it is used to manage and list features of the applications, prioritize the features,set the version ,release dates and key mile stones. • Power point presentations are used for planning meetings, as they help to put the feature details, related UI specifications,screenshots ,tasks all in on place and present it easily. • Accept360 is used just to store the details after planning ,it is only used to document the project execution details ,so that it is visible for all stakeholders involved in the mobile device,with restricted usage and permissions. • Accept360 is more good for big project on device level releases, to check that every component is available with required features confirming the requirements availability.It is not good for small projects like application level planning.
Project Situations	<ul style="list-style-type: none"> • What kind of deadlines are set in your team ? <i>Milestones / Releases/ feature completion?</i> • How are the deadlines achieved in the team? - <i>All ways successful or Some delay? Earlier failures or delays to meet deadlines? Please give more details on this.</i> 	<ul style="list-style-type: none"> • Team follows the organisational milestone process explained in the picture below ,Initially a business planning is done for a proposed application, where the requirements for the application is turned into the backlogs, this is done for a period of two months, after ADI ,the Project Initiation milestone is done ,after which the project is in the developmet stage where it is prototyped for around 3 months, after that execution milestone, the ADX is reviewed ,for which the project is evaluated to be continued or dropped, if the project is accepted to be continued, the ADX approval is done and the project then is executed for around 3 months ,after which the ADX milestone review is done,after that the product is published to be market.

		 <ul style="list-style-type: none"> • Deadlines are tied to the device releases and depends on the stability of them.
Project Reports	<ul style="list-style-type: none"> • What kind of reports are made to understand the project situations? • Does the Task board have all the info on what is done at any point of time. 	<ul style="list-style-type: none"> • Report and planning done in excel and ppt documents for sprint planning.
Suggetions	<ul style="list-style-type: none"> • Do you have any suggestion to improve the tracing and execution of the projects in the current organization ?in Nokia ?, • • What improvements would you like to have in the accept360 tool ? • Do you wish to have any improvemetns in the current way of usage of planning and execution tools to increase the productivity of the team? <i>Please give all your free thoughts that you feel might help improving the methods of project execution and that could help me define and create a proposal for the improvements in the tools</i> 	<ul style="list-style-type: none"> • Accept360 is not a fully web tool and needs plugin installation and works very slow,so it is better to have fully web tool,text based ,which can be used in all browsers and any operating systems, UI specification designing needs mac os and the inputs from mac UI specification needs to be affatched to the backlogs for following it up, this feature is missing in Accept360 . • Tool could provide support for contribution from anyone involved in the project ,to add notes/stories /suggestions / details , in the tool would be good. • If the task added can be supported by attachments of UI spec it will help a lot. • Tool that supports discussion to be added for the requirements. • Categorizing the Backlog items into groups would be good.(more details in the sketch in the ppt) • Accept360 is not helping for daily planning and adding notes, would be nice to have features available in other tools, for example from the experience the VersionOne tool from http://www.versionone.com/,was used in HERE projects was very good tool, it is a fully web based tool , good support for product planning, release palnning and sprint planning support for scrum projects_ http://www.versionone.com/ . • Scrumworks pro tool is was also good option.

		<ul style="list-style-type: none"> Desired features in Tools would look like in picture below 
--	--	---

Appendix 4. Interview

Date :-	15 January ,2014	
Role :-	Project Manager –Team X	
Project Details	<ul style="list-style-type: none"> What kind of work is done in the team? <i>e.g Development of Applications ? Services ? Quality enhancements?</i> 	<ul style="list-style-type: none"> Fully featured Application Development.
Team Info	<ul style="list-style-type: none"> What is the Team Size? What is the Team Structure ? 	<ul style="list-style-type: none"> 7 developers,Project Manager and Program Manager ,Test Lead + Testing team.
Agile Practices	<ul style="list-style-type: none"> What are the Agile practice followed in the team? scrum or kanban ? How often the requirements and the deadlines/milestones are planned/ updated? - <i>weekly planning? sprint planning ?</i> How are daily tasks planned /tracked/ updated? - <i>Daily Scrum ?</i> 	<ul style="list-style-type: none"> Scrum Method is Used,with 2 weeks Sprints One Day Sprint Review/ Planning meeting Project Manager heads the Planning Sprint meeting. Program manager heads the Sprint Review meeting. Daily Scrums are planned to happen daily, but mostly the team skips it, if the project manager is not available. Minimal estimation for the task is done in sprint planning.There are many methods for estimation ,like poker cards methods, but we do not use them as they are not very good and people do not give proper inputs for it. Kanban process is not tried in the team yet.
Project Planning and Execution Info	<ul style="list-style-type: none"> Team uses using accept360 tool ? How frequently the accept360 tool is used in the team? What are the other tools used in the team for tracking and execution of tasks? How do you use these tools for daily planning and updating the tasks? 	<ul style="list-style-type: none"> Initiation-Execution-Release Milestones method is used for the project execution. At Initiation stage, initially starts with an identified idea ,supported with the One pager Report, which explains the Key Features of the Idea and the Risks involved in developing the features.Next step is to collect all the

		<p>requirements and add to Accept360, next step is Flexibility study , which is prototyping the idea with possible features ,which ususally takes 2 weeks to 8 weeks.</p> <ul style="list-style-type: none"> • Next comes the Execution Commitment stage, the requirements need to be updated to the Accept360 ,before the project execution starts. • Requirements for the project has following hierarchy , Appication Name-> Folder for each Version->SubFeatures → User Story/ UseCases -> Tasks . Current Version is v.0. (Previous versions of the Application Y requirements do not have any standard templates that can be used for current version) • Each User Story has a description which has a Scenaria(Screen /UI)→ Action for it (What User Does)-> Outcome of the Action (Given a condition for action, what happens) • Project mangement team (Program manager,Project manager and Testing Team lead) have meetings to discuss the issues, this happens every week or even daily based on the situation of the project needs.Changes in features for implementaion are discussed during the meeting. • Nzilla and Perforce not used much. • Bugs are not included in Accept360. • Testing Tasks are also not included, but in case some unit testing is needed, then it will be considered. • A very good Definition of Done is defined, but using it for checking the completion at every 2 weeks sprint would be too much of useless effort put on developer's efficiency ,it would take more than two weeks to get a story done competely, but there is always some thing done and demo is shown at the review meeting. • Sprint iteration(e.g.4 sprints combined) are not used ,this could be used for checking the definition of done,as it gives more time for developers to finish the features completely, but it might endup in following to much of process and not helping to achieve good development .
--	--	---

Project Situations	<ul style="list-style-type: none"> • What kind of deadlines are set in your team ? <i>Milestones / Releases/ feature completion?</i> • How are the deadlines achieved in the team? - <i>All ways successful or Some delay? Earlier failures or delays to meet deadlines?</i> Please give more details on this. 	<ul style="list-style-type: none"> • Release milestones are set in the Accept360, it mainly depends on the deadlines of the product going to the market or any system release planned ,the applications go along with them. • A planned release date is set and 2 weeks buffer is also set before this date ,so that there is time to fix the issues found later.Also a secret Release date, later than the release date is the actual release date for the application.
Project Reports	<ul style="list-style-type: none"> • What kind of reports are made to understand the project situations? 	<ul style="list-style-type: none"> • Accept360 is used to assign the requirements to team, it is used to set planned dates for releases and milestones.From the information of these dates, the report is generated automatically from the tool in roger.nokia.com. • Separate project management checklist is maintained to check that the items , stories and contents in the Accept30 are upto date and valid.
Suggestions	<ul style="list-style-type: none"> • Do you have any suggestion to improve the tracing and execution of the projects in the organization ?, • What improvements would you like to have in the accept360 tool ? • Do you wish to have any improvements in the current way of usage of planning and execution tools to increase the productivity of the team? <i>Please give all your free thoughts that you feel might help improving the methods of project execution and that could help me define and create a proposal for the improvements in the tools</i> 	<ul style="list-style-type: none"> • Accept360 tool is too complex ,with too many features. • Mainly Accept360 is too complicated for the developers to use , with too much of information for them. • Risk Management features are missing in the tool. • UI is too slow, takes too much time to load, that is the reason most of the people are not actively using the tool.

Appendix 5. Interview

Date :-	20, January 2014	
Role :-	Lead Developer – Team X	
Project Details	<ul style="list-style-type: none"> • What kind of work is done in the team? <i>e.g Development of Applications ? Services ? Quality enhancements?</i> 	<ul style="list-style-type: none"> • Application Development
Team Info	<ul style="list-style-type: none"> • What is the Team Size? • What is the Team Structure ? 	<ul style="list-style-type: none"> • 7 developers,Project Manager and Program Manager ,Test Lead + Testing team

Agile Practices	<ul style="list-style-type: none"> • What are the Agile practice followed in the team? scrum or kanban ? • How often the requirements and the deadlines/milestones are planned/ updated? - <i>weekly planning? sprint planning ?</i> • How are daily tasks planned /tracked/ updated? - <i>Daily Scrum ?</i> 	<ul style="list-style-type: none"> • Scrum is used • Scrum in a light way,as it is the framework to adapt for our needs. • Business Owner /Program manager gives the requirements, discuss requiremetns with team and sets priority and keeps it optional for the team to pick them for the sprints. • Product owner/Project manager does most of the work of scrum master. • We don't follow retrospectives as it is waste of time. • Team decides what to do. • Only sprints are followed , no iterations are planned,but we do have reelases at some point which is kind of iterations, releases include some set of features. • Daily scrums are not important as we all sit close and discuss more often, and some one in the team skip it, so we stopped doing it. • Estimating is not in use, as it does not help much. • Hard task cannot be done by half of the team,so task defining not easy.When doing same kind tasks for long time, we get bored and take different task like UI ,for some time for change. • Team memebers pick task that they like to do. • Practice show that using burndown chart is not important.
Project Planning and Execution Info	<ul style="list-style-type: none"> • Team uses using accept360 tool ? • How frequently the accept360 tool is used in the team? • What are the other tools used in the team for tracking and execution of tasks? • How do you use these tools for daily planning and updating the tasks? 	<ul style="list-style-type: none"> • Team follows below mentioned mode of planning and execution. <ul style="list-style-type: none"> ◦ Design before implement like waterfall,Releasable product all the time. ◦ Think of essential things, just do it. ◦ Slice requirements into pices and do as much as possible. ◦ Initially we take more requirements and it is ok to take and see how that could be done. • Flow from Initiation Milestone to Execution Commitment milestone ,it is not the flow in scrum. • Changes do come in scrum sprints. • Scrumworks pro was good for daily use , it was used in previous project.

Project Situations	<ul style="list-style-type: none"> • What kind of deadlines are set in your team ? <i>Milestones / Releases/ feature completion?</i> • How are the deadlines achieved in the team? - <i>All ways successful or Some delay? Earlier failures or delays to meet deadlines?</i> Please give more details on this. 	<ul style="list-style-type: none"> • Deadlines are met well.
Project Reports	<ul style="list-style-type: none"> • What kind of reports are made to understand the project situations? 	Business owner and Project Owner makes reports,no reports from Lead developer.
Suggestions	<ul style="list-style-type: none"> • Do you have any suggestion to improve the tracing and execution of the projects in the current organization ? • What improvements would you like to have in the accept360 tool ? • Do you wish to have any improvements in the current way of usage of planning and execution tools to increase the productivity of the team? <i>Please give all your free thoughts that you feel might help improving the methods of project execution and that could help me define and create a proposal for the improvements in the tools</i> 	<ul style="list-style-type: none"> • Accept360 Tool is not useful for daily work planning. • Tool may be easy to use for planning/requirements managements, but not for daily execution of tasks • Since the tool has requirements and task in one place it is good, but it is slow. • Stories just say a feature, do not specify any UI details,this is not good. • Can't add Task to requirements, we need to add dummy stories which are just tasks , that is not very good. • Requirements could be specified only very small,but actual implementation for it is quit big. Beter ways might be good.

Appendix 6. Interview

Date :-	11,February 2014	
Role :-	Quality Lead	
Project Details	<ul style="list-style-type: none"> • What kind of work is done in the team? <i>e.g Development of Applicatioans ? Services ? Quality enhancements?</i> 	<ul style="list-style-type: none"> • Incremental release of Application Y with some feature addition.
Team Info	<ul style="list-style-type: none"> • What is the Team Size? • What is the Team Structure ? • How many Teams ? • If Teams are located at different location .Where are the Teams located? How do they share work? 	<ul style="list-style-type: none"> • Team has 3 developers and one tester. • Team is located in Espoo, one team
Agile Practices	<ul style="list-style-type: none"> • What are the Agile practice followed in the team? scrum or kanban ? 	<ul style="list-style-type: none"> • Scrum is used. • Everyday meeting is held to recap on the things done and next plan ,with the status checks.

	<ul style="list-style-type: none"> • How often the requirements and the deadlines/milestones are planned/ updated? - <i>weekly planning? sprint planning ?</i> • How are daily tasks planned /tracked/ updated? - <i>Daily Scrum ?</i> 	<ul style="list-style-type: none"> • Team follows 2 Weeks Sprint. • Releases for testing and the sprint depends on the priority of Release Milestone schedule and updates to be available in store. • Incremental release of the application some times only have bug fixing and store update.
Project Planning and Execution Info	<ul style="list-style-type: none"> • How are the requirements planned ? • How often the requirements are updated or prioritised ? • Team uses using accept360 tool ? • How frequently the accept360 tool is used in the team? • What are the other tools used in the team for tracking and execution of tasks? • How do you use these tools for daily planning and updating the tasks? 	<ul style="list-style-type: none"> • At Project Initiation Milestone ,only information about the testing contacts is planned ,test process is already available or being planned at this stage. • There are no other activities involved in the initiation milestone stage ,only the future plan is communicated. • Initiation Milestone: At this stage there is alpha release of the application and automation goals should be set for that ,in Execution commitment milestone, meaning the required tools and resources has to be identified. • Perforce(P4) is used,but P4 is not ready by Execution commitment milestone, only after that the application resources are available in P4. • We follow L1, L2 and L3 level releases, for integration purpose,and consider Execution commitment and Release milestones as program milestones. • When a build is decided to be store candidate or Release Milestone candidate, it has to be promoted to L1. • Basically the git source code changes might be available in P4 at L3,but testing would only fetch from L2 level. • The changes committed in L3 would be cherry picked or based on the requirement or sprint plan, a new release would be made and pushed to L2 • A full test cycle is performed on the release or xap from L2 • For L3 level release, Feature level testing and bug reporting is done. • For L2 level release, Pre-Integration testing based on sprint activities is done. • Capabiltiy to run the application in all devices is done before L1. • All L3 candidates are final approved candidates

		<ul style="list-style-type: none"> L3 could be on every sprint or every day. If the sprint release has many critical bugs, then it has to be fixed within the next sprint L3.
Project Situations	<ul style="list-style-type: none"> What kind of deadlines are set in your team? <i>Milestones / Releases/ feature completion?</i> How are the deadlines achieved in the team? - <i>All ways successful or Some delay? Earlier failures or delays to meet deadlines?</i> Please give more details on this. 	<ul style="list-style-type: none"> Release Milestone is followed, for which ADR template is used, which requires Functional Testing (FT) report to be good and, No critical and show stopper bugs, other bugs could be there. After Release Milestone, Certification testing and Non-Functional Testing is done.
Project Reports	<ul style="list-style-type: none"> What kind of reports are made to understand the project situations? Does the Task board have all the info on what is done at any point of time. 	<ul style="list-style-type: none"> At L2 daily test reports are created, bugs are reported in nzilla. For FT Weekly reporting, synchronization with Quality center and generating report from it is done. Pre-Certification test report. NFT report LVT sanity test report
Suggestions	<ul style="list-style-type: none"> Do you have any suggestion to improve the tracing and execution of the projects in the current organization? What improvements would you like to have in the accept360 tool? Do you wish to have any improvements in the current way of usage of planning and execution tools to increase the productivity of the team? <p><i>Please give all your free thoughts that you feel might help improving the methods of project execution and that could help me define and create a proposal for the improvements in the tools</i></p>	<ul style="list-style-type: none"> Accept360 it only has tasks which are specific, but in reality there are many tasks done, but are not reflected as tasks in Accept360. For example there is only one task in A360 as for testing, but there are many testing tasks done, also we do get some extra tasks, if there is new libraries added, that need to be tried and we are involved by these kind of tasks are not reflected anywhere in the tool. Also there is only one task added and only one person name is assigned to it, but many others are also working on the different task for the same product. If we can have more details in the tool, it will be good.

Appendix 7. Interview

Date :-	29, January, 2014	
Role :-	Lead Program Manager -Team Y	
Project Details	<ul style="list-style-type: none"> What kind of work is done in the team? <i>e.g Development of Applications ? Services ? Quality enhancements?</i> 	<ul style="list-style-type: none"> Application, Driver and SDK Development
Team Info	<ul style="list-style-type: none"> What is the Team Size? What is the Team Structure? 	<ul style="list-style-type: none"> Teams are located in three different sites. Program Manager Works on creating requirements and features with many teams on daily basis.

		<ul style="list-style-type: none"> Teams are structured on the type of components they create, Application development team, which dependencies with to other software (SDKs) and hardware (drivers) components.
Agile Practices	<ul style="list-style-type: none"> What are the Agile practice followed in the team? scrum or kanban ? How often the requirements and the deadlines/milestones are planned/ updated? - <i>weekly planning? sprint planning ?</i> How are daily tasks planned /tracked/ updated? - <i>Daily Scrum ?</i> 	<ul style="list-style-type: none"> Scrum is followed Do not use more time for estimation, as it doesn't turn out accurate and not very useful.
Project Planning and Execution Info	<ul style="list-style-type: none"> Team uses using accept360 tool ? How frequently the accept360 tool is used in the team? What are the other tools used in the team for tracking and execution of tasks? How do you use these tools for daily planning and updating the tasks? 	<ul style="list-style-type: none"> Team follows many project Initiation milestones, within the actual Initiation Milestone with small releases. In Site A program manager feels that the processes followed are more like waterfall model and less agile. Accept360 is more like documenting tool ,for putting in the details what we do ,so that everybody located in different place, could view and know the situations, it does not help much in planning. Accept360 is not very good for day to day planning ,so another tool is needed for this. In Site B they use the tool JIRA for scrum planning and execution. It helps to follow milestones, iterations and bug handling. https://www.atlassian.com/software/jira In Site C they do follow different practices of scrum and tools, Site C lead could be contacted for more details if needed. Continuous integration of features and weekly build are used and is good for getting the requirements done. Burndown chart are good only when we are doing bug fixing works, but it is not good for feature development phases.
Project Situations	<ul style="list-style-type: none"> What kind of deadlines are set in your team ? <i>Milestones / Releases/ feature completion?</i> How are the deadlines achieved in the team? - <i>All ways successful or Some delay? Earlier failures or delays to meet</i> 	<ul style="list-style-type: none"> Deadlines are timeboxed to the market releases of the devices ,it is ok that we keep our deliverable deadlines tied to it. But Deadlines are tightly tied with the device release and readiness, so mostly when the project is planned to be released ,it is ready, but

	<p><i>deadlines?</i> Please give more details on this.</p>	<p>then we get the new device at that time and devices is not stable enough and then team has to work with it for one month, to get the bugs related to the device softwares fixed.</p> <ul style="list-style-type: none"> There is no tool that help to relate the dependencies with different components like system software ,hardware drivers , application etc. Accept360 does not have this dependency tied up in one place.
fProject Reports	<ul style="list-style-type: none"> What kind of reports are made to understand the project situations? 	<ul style="list-style-type: none"> Personally follow up with the plans and reports as there are teams from different location and dependencies between them and between different components .
Suggetions	<ul style="list-style-type: none"> Do you have any suggestion to improve the tracing and execution of the projects in the current organization ? What improvements would you like to have in the accept360 tool ? Do you wish to have any improvemetns in the current way of usage of planning and execution tools to increase the productivity of the team? <p><i>Please give all your free thoughts that you feel might help improving the methods of project execution and that could help me define and create a proposal for the improvements in the tools</i></p>	<ul style="list-style-type: none"> Features in the tool to have the project dependencies available at one tool/place and easy to use by everyone to add more details . Involving Engineering tasks in initial stage of ADI (Initiation Milestone) would be good and having small ADI is good and tool might be good,if it is has this kind of support for planning,the picture below is the sketch from the discussion board.

Appendix 8. Interview

Date :-	11 ,February 2014
Role :-	Project Manager – Team Y

Project Details	<ul style="list-style-type: none"> What kind of work is done in the team? <i>e.g Development of Applicatioans ? Services ? Quality enhancements?</i> 	<ul style="list-style-type: none"> Development of enhancement features for existing application X and porting to the new platforms.
Team Info	<ul style="list-style-type: none"> What is the Team Size? What is the Team Structure ? How many Teams ? If Teams are located at different location .Where are the Teams located? How do they share work? 	<ul style="list-style-type: none"> 20-30 developers located at 3 different sites
Agile Practices	<ul style="list-style-type: none"> What are the Agile practice followed in the team? scrum or kanban ? How often the requirements and the deadlines/milestones are planned/ updated? - <i>weekly planning? sprint planning ?</i> How are daily tasks planned /tracked/ updated? - <i>Daily Scrum ?</i> 	<ul style="list-style-type: none"> Scrum is followed . Weekly planning is usually used, but if needed daily meeting and Adhoc meetings are planned. Daily tasks are planned in OneNote and Adhoc meetings. We follow scrum in a adapted way, if things don't work we change the way and adopt, we don't follow books.
Project Planning and Execution Info	<ul style="list-style-type: none"> How are the requirements planned ? How often the requirements are updated or prioritised ? Team uses using accept360 tool ? How frequently the accept360 tool is used in the team? What are the other tools used in the team for tracking and execution of tasks? How do you use these tools for daily planning and updating the tasks? 	<ul style="list-style-type: none"> Requirements are listed in OneNote,those are list of features needed in the application. We implement featurres and see if it is completed in the planned time, if compelte then we release else move to next release. We don't use Accept 360. Product management uses A360 ,Program manager is suppose to update the data in Accept360, but the current features and details in A360 does not seems to be updated for current feature in development. Use Only OneNote for all planning and execution, it is very helpful
Project Situations	<ul style="list-style-type: none"> What kind of deadlines are set in your team ? <i>Milestones / Releases/ feature completion?</i> How are the deadlines achieved in the team? - <i>All ways successful or Some delay? Earlier failures or delays to meet deadlines? Please give more details on this.</i> 	<ul style="list-style-type: none"> There are releases planned for one month , for example , releases planned in next short periods are on week 18,23, 38.. We meet deadlines.No Delays
Project Reports	<ul style="list-style-type: none"> What kind of reports are made to understand the project situations? Does the Task board have all the info on what is done at any point of time. 	<ul style="list-style-type: none"> We do reports and status update in OneNote Page,for example, a list with features planned date status is available in OneNote We use only OneNote Page and it has all information,

Suggestions	<ul style="list-style-type: none"> Do you have any suggestion to improve the tracing and execution of the projects in the current organization ? What improvements would you like to have in the accept360 tool ? Do you wish to have any improvements in the current way of usage of planning and execution tools to increase the productivity of the team? <p><i>Please give all your free thoughts that you feel might help improving the methods of project execution and that could help me define and create a proposal for the improvements in the tools</i></p>	<ul style="list-style-type: none"> Accept360 is not useful, it needs to be replaced by some other tool.
-------------	--	--

Appendix 9. Interview

Date :-	6 ,March 2014	
Role :-	Lead Developer - Team Y	
Project Details	<ul style="list-style-type: none"> What kind of work is done in the team? <i>e.g Development of Applications ? Services ? Quality enhancements?</i> 	<ul style="list-style-type: none"> Application Development.
Team Info	<ul style="list-style-type: none"> What is the Team Size? What is the Team Structure ? How many Teams ? If Teams are located at different location .Where are the Teams located? How do they share work? 	<ul style="list-style-type: none"> 10 Engineers ,4 QA people Teams are structured based on the work/task Teams are in three different sites
Agile Practices	<ul style="list-style-type: none"> What are the Agile practice followed in the team? scrum or kanban ? How often the requirements and the deadlines/milestones are planned/ updated? - <i>weekly planning? sprint planning ?</i> How are daily tasks planned /tracked/ updated? - <i>Daily Scrum ?</i> 	<ul style="list-style-type: none"> Scrum is followed . Scrum with weekly Sprint planning meetings. Adhoc meetings when ever needed. All Friday afternoon Meeting with free agenda, Kind of Retrospective meeting and Technical catchup meeting. Daily tasks are planned in Adhoc meetings, whenever needed. Lead developer has to check the progress and talk to people to communicate task details.
Project Planning and Execution Info	<ul style="list-style-type: none"> How are the requirements planned ? How often the requirements are updated or prioritised ? Team uses using accept360 tool ? 	<ul style="list-style-type: none"> Project Backlogs(BL) are available in OneNote, Product/Project Manger / Business Owner updates it. Backlogs are high level requirements.

	<ul style="list-style-type: none"> • How frequently the accept360 tool is used in the team? • What are the other tools used in the team for tracking and execution of tasks? • How do you use these tools for daily planning and updating the tasks? 	<ul style="list-style-type: none"> • PM/Director/UX/Architects are involved in planning the BLs. • Product managers does and communicates the major changes. • List of BLs for next 6 months or 1 year is planned and available. • A copy of selected BLs are the considered tasks. • The BL are prototyped first and then implemented completely. • UI specs come in randomly and are changed when ever needed. • OneNote is used for all planning and execution. • Nzilla is used. Not using Perforce. • Scrum fails because the developers don't update their status themselves. • Features are planned to be completed for the store releases. • Weekly releases for testing team. • Push to Perforce once when needed, git is used more . • Project Initiation Milestone → ProtoTyping, Investigating tasks. • Create different versions of application to be released to the store. • Execution commitment milestone → Implementations with no bugs and with PM feedback as OK ,the application is released to Store (Release Milestone).
Project Situations	<ul style="list-style-type: none"> • What kind of deadlines are set in your team ? <i>Milestones / Releases/ feature completion?</i> • How are the deadlines achieved in the team? - <i>All ways successful or Some delay? Earlier failures or delays to meet deadlines?</i> Please give more details on this. 	<ul style="list-style-type: none"> • Always successful, No delays. • Quality and Time is our concern, features are reduced/removed when no time is there to implement or if quality is bad .
Project Reports	<ul style="list-style-type: none"> • What kind of reports are made to understand the project situations? • Does the Task board have all the info on what is done at any point of time. 	<ul style="list-style-type: none"> • Task status are understood from the git commits, the lead picks up the information from the git commit messages and puts it in the OneNote and makes the reports. • Developers are suppose to update the daily tasks in OneNote page with status DONE, but mostly people don't update this ,so lead has to follow up and update it.

Suggetions	<ul style="list-style-type: none"> Do you have any suggestion to improve the tracing and execution of the projects in the current organization ? What improvements would you like to have in the accept360 tool ? Do you wish to have any improvemetns in the current way of usage of planning and execution tools to increase the productivity of the team? <p><i>Please give all your free thoughts that you feel might help improving the methods of project execution and that could help me define and create a proposal for the improvements in the tools</i></p>	<ul style="list-style-type: none"> A360 is very slow and cannot be used at all. It is way too complicated. The tool need to be rewritten from the scratch to be easier to use, it has too many buttons, too many sub-typings. We need only a simple tool to list the things to do.
------------	--	--

Appendix 10. Interview

Date :-	23 ,April 2014	
Role :-	QA Lead – Team Y	
Project Details	<ul style="list-style-type: none"> What kind of work is done in the team? <i>e.g Development of Applicatioans ? Services ? Quality enhancements?</i> 	<ul style="list-style-type: none"> Maintenace and feature addition to the application.
Team Info	<ul style="list-style-type: none"> What is the Team Size? What is the Team Structure ? How many Teams ? If Teams are located at different location .Where are the Teams located? How do they share work? 	<ul style="list-style-type: none"> QA 2 leads, 3 testers, there are small tems inside the Team C , so the QA people are allocated as and when needed for teams.
Agile Practices	<ul style="list-style-type: none"> What are the Agile practice followed in the team? scrum or kanban ? How often the requirements and the deadlines/milestones are planned/ updated? - <i>weekly planning? sprint planning ?</i> How are daily tasks planned /tracked/ updated? - <i>Daily Scrum ?</i> 	<ul style="list-style-type: none"> Currently no daily scrums, before there used to be daily scrums. Scope is small and we meet with developers and plan the testing. Weekly planing meetings. Mosly bug fixing activities and planning is done using the One Note tool and activities are checked from it.
Project Planning and Execution Info	<ul style="list-style-type: none"> How are the requirements planned ? How often the requirements are updated or prioritised ? Team uses using accept360 tool ? How frequently the accept360 tool is used in the team? 	<ul style="list-style-type: none"> Requirement and functionality information is available in UI design documents,it has most of the information we need for testing. some times user stories in the A360 could be used.But stories are not update always Test plans document is done with a stndard template and available in sharepoint ,but this is

	<ul style="list-style-type: none"> • What are the other tools used in the team for tracking and execution of tasks? • How do you use these tools for daily planning and updating the tasks? 	<p>more generic. Testing team uses test cases more for checking the functionality, test cases are available in QC and we add, obsolete the according to the requirements of the application.</p> <ul style="list-style-type: none"> • One Note is mainly used for project reports. • List of bugs that need to be worked and the list from nzilla and its link is available in one note page of the project, and updated when the meetings happen. • Perforce, nzilla, QC are the tools used. • Accept360 is not used more. • We prioritize the bugs daily, Project manager and QA lead meet to update this list. • L1 – Release done within team. • L2 – Release of XAP file for more audience. • L3 – Release to the market.
Project Situations	<ul style="list-style-type: none"> • What kind of deadlines are set in your team? <i>Milestones / Releases/ feature completion?</i> • How are the deadlines achieved in the team? - <i>All ways successful or Some delay? Earlier failures or delays to meet deadlines?</i> Please give more details on this. 	<ul style="list-style-type: none"> • Team is always close to deadline and bugs are fixed on time.
Project Reports	<ul style="list-style-type: none"> • What kind of reports are made to understand the project situations? • Does the Task board have all the info on what is done at any point of time. 	<ul style="list-style-type: none"> • WRT tool is used for creating bug reports.
Suggestions	<ul style="list-style-type: none"> • Do you have any suggestion to improve the tracing and execution of the projects in the current organization? • What improvements would you like to have in the accept360 tool? • Do you wish to have any improvements in the current way of usage of planning and execution tools to increase the productivity of the team? <p><i>Please give all your free thoughts that you feel might help improving the methods of project execution and that could help me define and create a proposal for the improvements in the tools</i></p>	<ul style="list-style-type: none"> • A360 is not used more, but it is not being passed, but it is updated when needed. • Face to face meetings are enough to explain the problems and bugs that helps, so we do not use other tools much, hence no suggestions.