



VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES

Victorien Alric

# TASK MANAGEMENT IN A LARGE- SCALE ENVIRONMENT

Technology and Communication  
2021

## **ACKNOWLEDGMENT**

I would like to express my sincere gratitude to my supervisor Dr. Ghodrat Moghadampour and Ms. Sanna-Liisa Kiikkala for their valuable advice and continuous support during the thesis elaboration.

## ABSTRACT

Author	Victorien Alric
Title	Task Management in a Large-scale Environment
Year	2021
Language	English
Pages	29
Name of Supervisor	Ghodrat Moghadampour

---

This document describes the results of a task management project in a large-scale environment. The project was initiated by a customer of Almnorth Oy and consisted of unifying task management process in a large scale, involving several hundreds of persons. Almnorth Oy is a Vaasa based company providing consulting services in Application Lifecycle Management (ALM).

Large companies are organized in several departments that might have their own ways of working; the aim of this project was to unify the way of working across selected departments. Doing so enables traceability and transparency of ongoing activities in each department, resulting in a better visibility of available workforce and enabling management teams to make decision to allocate resources where needed and provides a general overview of activities within selected departments.

The thesis proposes an implementation example of a task management system using Polarion ALM software and highlights key elements of the tool, such as reporting and advanced configuration. This case study has been standardised and simplified so it can be applied in most enterprises, regardless of their working methodology.

All in all, this task management project provided satisfying results by enabling transparency and facilitating communication between employees among selected departments.

---

Keywords	Polarion, Application Lifecycle Management, enterprise management, work methods and task management
----------	---

# CONTENTS

ACKNOWLEDGMENT

ABSTRACT

1	INTRODUCTION .....	1
1.1	Bases of Project Management.....	1
1.2	Project Background.....	1
1.3	Project Objectives .....	2
1.4	Project Scope .....	3
2	TECHNOLOGIES AND TOOLS .....	4
3	TECHNICAL BACKGROUND INFORMATION .....	6
3.1	Key Polarion Features.....	6
4	TEAMS AND ROLES .....	9
5	PROJECT PHASES .....	10
5.1	Project Planning .....	10
5.2	Work Breakdown .....	11
5.3	Work Planning .....	11
5.4	Work Execution .....	11
6	DEFINITION OF WORK ITEM TYPES .....	12
6.1	Project Package.....	13
6.2	Work Request.....	16
6.3	Tasks .....	17
7	POLARION ADVANCED SCRIPTING.....	20
7.1	Workflow scripting .....	20
7.2	Highcharts Live Report Development .....	22
8	CONCLUSIONS .....	28
	REFERENCES.....	29

## List of abbreviations

<b>ALM</b>	Application Lifecycle Management
<b>HTML</b>	Hypertext Markup Language
<b>LDAP</b>	Lightweight Directory Access Protocol
<b>SVN</b>	Apache Subversion – File versioning system
<b>XML</b>	Extensible Markup Language
<b>JSON</b>	JavaScript Object Notation
<b>API</b>	Application Programming Interface
<b>UI</b>	User Interface
<b>SOAP</b>	Simple Object Access Protocol
<b>WSDL</b>	Web Services Description Language

## LIST OF FIGURES AND TABLES

<b>Figure 1.</b> ALM Circle (Polarion) .....	2
<b>Figure 2.</b> Project phases diagram and corresponding Polarion workItems .....	10
<b>Figure 3.</b> Polarion "Form Layout Configuration" example for Work Request....	13
<b>Figure 4.</b> Display interface of Project Package workItem .....	14
<b>Figure 5.</b> Workflow configuration overview for Project Package workItem.....	14
<b>Figure 6.</b> Initial workflow transitions of Project Package workItem .....	15
<b>Figure 7.</b> Linked item creation shortcut menu from a Project Package item .....	15
<b>Figure 8.</b> Display interface of Work Request workItem .....	16
<b>Figure 9.</b> Workflow configuration for Work Request workItem .....	17
<b>Figure 10.</b> Display interface of Task workItem .....	18
<b>Figure 11.</b> Sample workflow configuration for Task workItem .....	19
<b>Figure 12.</b> Workflow configuration for "Progress" action in Task workflow .....	22
<b>Figure 13</b> Workload vs Capacity Highcharts chart built from Polarion data.....	23

# **1 INTRODUCTION**

## **1.1 Bases of Project Management**

Project management is the process of organizing the work of a project with given constraints. The most common challenges in project management are budget, schedule planning and resources availability. The role of a project manager is to find balance between these parameters and carry the project to completion in accordance with initial requirements and scope.

A traditional project typically goes through the following phases: initialisation, planning, execution, monitoring and completion.

During the initialisation phase, the requirements are gathered, the final scope is defined and approved by relevant stakeholders. The planning phase is where the work is broken down in smaller pieces and the schedule is defined and communicated with the resources allocated for the project. The work is being performed by assigned teams during the execution phase. During the monitoring phase, the results are tested and verified against initial requirements to ensure the result is satisfying. Some small adjustments or refinement can be made before closing the project.

## **1.2 Project Background**

Almnorth is a company founded in 2019, who provides Application Lifecycle Management consulting services with the help of the Polarion ALM software from Siemens.

The studied task management project has been conducted by a customer of Almnorth and have been successfully deployed throughout 2019.



**Figure 1.** ALM Circle (Polarion)

Application lifecycle management is a framework through which the process of software development and ongoing maintenance is recorded and controlled. The ALM methodology is a more comprehensive approach than the traditional systems development lifecycle because it includes continuous management from an inception of the application until its decommissioning. There may be multiple systems development lifecycles throughout the life of an application product.

### 1.3 Project Objectives

The main purpose of the studied task management project is to have a unified way of working and a common tool for all employees in selected departments, enabling traceability of the work taking place in between these departments, increasing visibility and predictability of upcoming activities while facilitating the prioritization process. An efficient task management system also speeds up the information sharing of dependent work from a department to another by automating communication of key information.



## **1.4 Project Scope**

The scope of this project is limited to the task management and project management aspect and does not cover requirements management or product lifecycle management. It is a more generic approach that may apply to a variety of enterprises. One of the goals is to demonstrate capabilities and flexibility of the Polarion software and demonstrate how the tool can be tailored for demanding environments.

## 2 TECHNOLOGIES AND TOOLS

This section lists all technologies used for implementing the task management solution proposed.

- Polarion ALM

Polarion is a tool that hosts all data and configuration related to the task management project. Polarion has the ability to define “workItems” which are customer defined objects containing pre-defined fields. Polarion stores all workItems in the XML format in an integrated Subversion repository and provides a browser-based user-interface to create, configure and display workItems. /1/

- Apache Velocity v1.4

Apache Velocity (v1.4) is a language used by Polarion for internal scripts or reports. It is available in “Rich Pages” or “Live Report Pages” which allow customers to write their own scripts.

- JavaScript

Polarion supports the JavaScript language in “Live Report Pages” alongside the HTML code. JavaScript can also be used in scripts hosted on the Polarion server, for example workflow scripts or scheduled scripts.

- Highcharts

Highcharts is a JavaScript based library whose purpose is to represent JSON data in various chart formats. The Highcharts library is included in Polarion which provides a widget to assist the configuration of basic Highcharts charts, while the complete Highcharts library remain available through Polarion scripting where charts can be customized entirely from this interface. Highcharts is a well-documented library /5/.

- Apache Lucene

Apache Lucene is a simple query language used by Polarion to make workItems queries and browse through Polarion's database. /6/

The following query example can be used in Polarion to fetch workItems of type "Task" assigned to user with id "U001" in "Open" status.

```
"type:task AND assignee:U001 AND status:open"
```

### 3 TECHNICAL BACKGROUND INFORMATION

Polarion ALM was first released in 2004 by Polarion Software and acquired by Siemens in January 2016. Polarion is a unified solution for Requirements, Quality, and Application Lifecycle Management. Targeted at large enterprises, it allows any given organization to deeply customise the software to meet complex and specific requirements. /1/

Polarion ALM gives organizations one unified solution that delivers project transparency through real-time aggregated management information. Everyone is aligned around what is being built and why to drive advancement while protecting integrity and compliance. This helps teams respond faster and with better quality to new business opportunities and customer demands. The solution is based around three core pillars: collaboration, traceability and reuse of content.

#### 3.1 Key Polarion Features

- User management

Like the majority of productivity tools, Polarion needs a system to manage its users. Each user has an advanced set of permission applied to specific projects. The Polarion user database can be integrated with well-known identity management systems, such as OAuth, Kubernetes or LDAP.

- Projects

A Polarion project is a space within the repository, similar to a file folder. All configuration and content related to a project is stored within the project folder. The project has its own Polarion configuration and its own set of users and permissions. Each project has its own configurable home page.

- WorkItems

A WorkItem is a representation of an object within Polarion. WorkItems are fully customizable. There is a number of predefined fields for all types of workItem but

it is up to the administrators to define different workItem types and select fields that shall be used and when needed, create additional fields.

- Documents

Documents are similar to “Microsoft Office Word” documents but allows creation of workItems by selecting relevant text and marking it as a workItem. These documents can be exported as PDF. A typical use of documents is to maintain and develop a list of requirements.

- Live Report Pages

Live Report Pages or Rich Pages, are live documents that can load widgets and run custom Apache Velocity scripts that supports HTML and JavaScript content as well. For example, the home page of each project consists of a Live Report Page and usually contains information on the project users and recent activity and a dynamic table of the workItems recently updated in the project.

- Plans

Plans are workItem collectors with a display page similar to a Live Report Page. A plan requires a starting date and ending date, it has its own workflow. For example, in software development projects, a plan might represent a two-weeks sprint and contain several WorkItems of type “Task” which are assigned to different persons. Polarion Plan is a great solution to enable a team to follow up progress on the scheduled tasks and provides a report that can be used during regular meetings held by the team.

- WorkItem linking

The workItem linking functionality allows workItems to be linked to another, regardless if they are in the same project. The type of link, referred to as “Link Role” is defined by the customer and is used to differentiate relations between items, “processes” and “depends on” are commonly used link role types.

All of the above mentioned Polarion core functionalities are available to users via a web browser through the Polarion user-interface, but some of the functionalities can be accessed through the Polarion API.

Polarion scripting interfaces are divided into three main APIs:

- Polarion Open Java API: This API is available in workflow scripts; scheduled scripts and some interfaces are available in Live Report Pages. /3/
- Polarion Rendering API: This API is mainly available in Live Report Pages, it is a read-only API that provides easy functions to render workItems and their fields. /4/
- Polarion Web Services API: This API is used to integrate Polarion with other tools using SOAP calls via the WSDL protocol.

These APIs are accessed in different manners from Polarion:

- Live Report pages provides read-only access to Polarion Rendering API to enable data retrieving and scripting directly inside Polarion.
- JavaScript scripts can access the Polarion Open API to make modifications to workItems or documents. These scripts can be loaded via Workflow trigger or via the Polarion job scheduler functionality.
- Classic Wiki pages can access the Polarion Open API as well as the Rendering API to either render content or modify content.
- An optional module called the “Polarion Advanced Scripting Engine” is available and allows deeper customization of Polarion, for example it is possible to trigger JavaScript code whenever a workItem is saved in Polarion.

## 4 TEAMS AND ROLES

In a large-scale enterprise, employees are assigned different roles and teams, they might work on the same project with different roles. A breakdown of different roles relevant for task management is detailed below.

- Project Managers

Project managers are responsible for the development of a given project, their role is to ensure the project meets the requirement and is completed within cost and time restrictions. To do so, the project managers have to break down and list all the work that needs to be done to fulfill project requirements. Once this step is achieved, they can provide an estimate for remaining cost and give a target completion date.

- Department Managers

Department managers oversee an entire department, their role is to ensure that operations are running smoothly within the department and manage resources. They are responsible for time allocation between different projects. Strong coordination between department managers, project managers and team managers are required to optimize performance.

- Team Managers

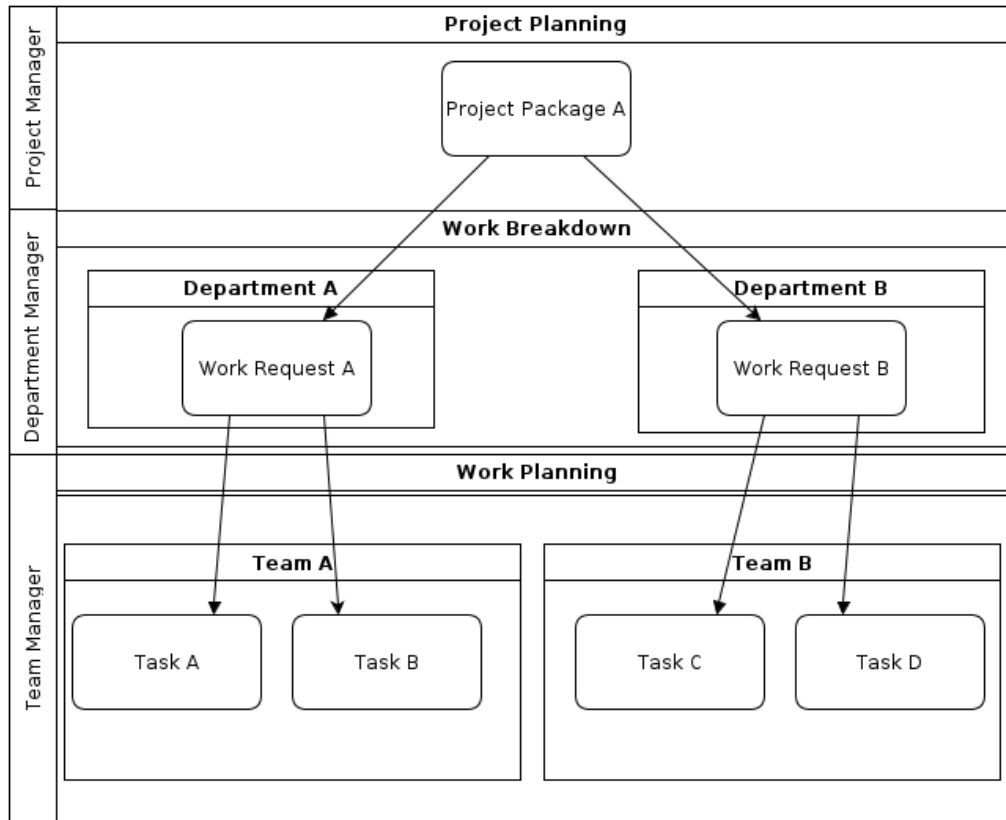
Team managers are responsible for a certain team. They plan the work together with the team members according to priorities coming from the department managers.

- Team Members

Team members are the persons executing tasks according to the schedule agreed with the team manager. It is essential to maintain strong coordination between members and organize regular meeting to ensure targeted schedules are met.

## 5 PROJECT PHASES

This chapter introduces the four main project phases in project management and their translation in Polarion configuration.



**Figure 2.** Project phases diagram and corresponding Polarion workItems

### 5.1 Project Planning

Project planning is the initial project phase which consists of deciding whether or not to create a new product or a new feature. An analysis is performed based on requirements handed over by customers or by internal stakeholders. During this phase, requirements are set, the initial project plan is prepared, schedule and costs are estimated. The work is broken down into Polarion workItems of type “Project Packages” and relevant departments are consulted to discuss the scope of the department engagement and commitment to the project. Scheduling is organized according to each department’s availability. This step will result in a project plan



approved by all interested parties. A “Project Package” can represent a large amount of work and is typically completed within months.

## **5.2 Work Breakdown**

During work breakdown, project managers and departments manager discuss and define Polarion workItems of type “Work Request” which are assigned to relevant departments with agreed schedules. A “Work Request” shall be completed within a few weeks so the “Project Package” is updated regularly.

Department managers discuss priorities and further distribution during regular order intake meetings.

## **5.3 Work Planning**

The team manager and team members gather to break own the work further from “Work Requests” to Polarion workItems of type “Task”. They prioritize and plan the work. Tasks are assigned to each team member according to their capacity and specialty.

## **5.4 Work Execution**

Once the work is planned, it is time to execute the work and complete the tasks. Once all tasks are completed, the parent “Work Request” item is set to “Closed” status. Once all “Work Requests” are closed, the parent “Project Package” can be set to “Completed” status. This process enables traceability across departments and the information displayed is always up to date. At the team level, the work is planned regularly and new tasks are processed within short time, a task shall not represent more than a few days of work.

## **6 DEFINITION OF WORK ITEM TYPES**

At the start of the project, the departments involved had different way of working with different tools. Some were sharing progress via email or tracking tasks via Excel while others had different task management tools.

The work breakdown process is different for each department but eventually the source of the work is common to all: business driven requirements which translate into a new project being initiated.

In this example, only three type of workItems are used, additional layers can be added according to organization structure, complexity and size.

Polarion has built-in fields for all item types which are enabled by default. These fields include: title, description, status, resolution, author, assignee, initial estimate, time spent, priority and comments. In addition to these fields, one may create additional custom fields of certain data type: single line text, html text, integer, floating point, date, Boolean or enumeration. It allows the work item objects to suit most of one's needs.

Once the fields are defined, Polarion offers workItem form layout customization settings in the form of an XML file and determine how the fields will be displayed when opening a workItem in a web browser.

## Form Layout Configuration

View: Default

Work Item Type: Work Request

Save Cancel

**Configuration has been saved.**

```

</vertical>
<vertical>
  <section>
    <field id="assignees"/>
    <field id="status"/>
    <field id="resolution"/>
    <field id="ROUGH_ESTIMATE"/>
    <field id="ActualSTARTDATE"/>
  </section>
</vertical>
<vertical>
  <section>
    <field id="COST_CENTER"/>
  </section>
  <section>
    <field id="STARTDATE"/>
    <field id="ENDDATE"/>
  </section>
</vertical>
<vertical>
  <section>
    <field id="type"/>
    <field id="project"/>
    <field id="authors"/>
    <field id="submittedBy"/>
  </section>
</vertical>

```

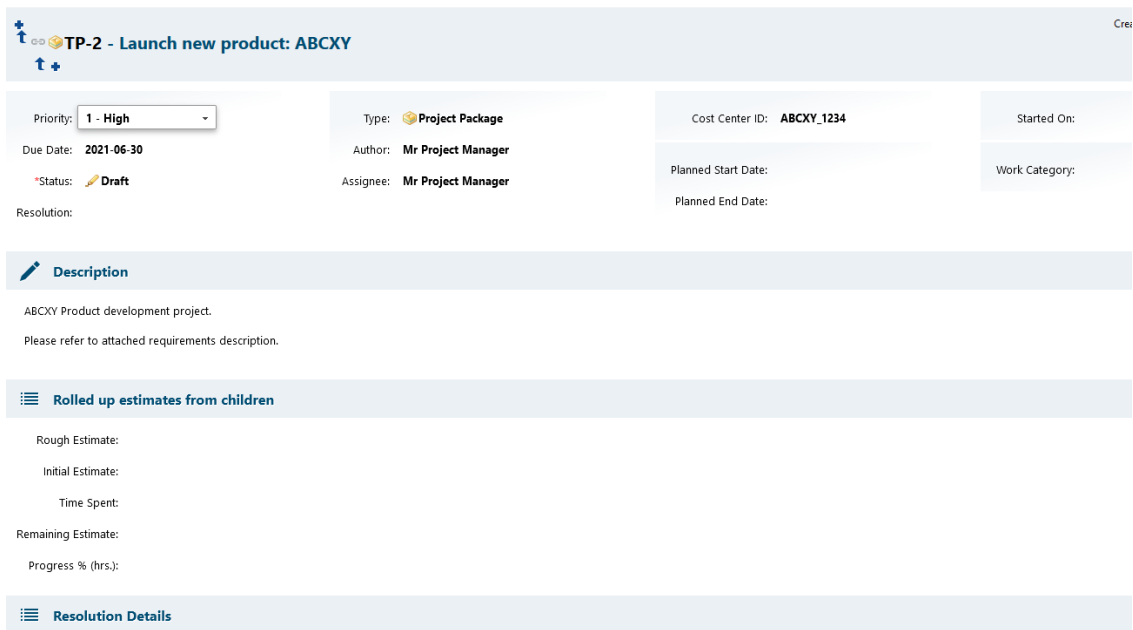
**Figure 3.** Polaron "Form Layout Configuration" example for Work Request

### 6.1 Project Package

The “Project Package” workItem is the highest “work collector” and is driven by the project management team.

Project Package workItem type uses the following fields:

- Standard fields: Title, Description, Due Date, Status, Priority, Comments
- Custom fields: Cost Center, Resolution Details
- Automatic fields: Planned Dates, Started On, Time Spent, Estimates
  - Automatic fields are calculated based on the sum of all children items of type “Work Request”. For example, Planned Start date comes from the earliest Planned Start Date of any of the children “Work Request”.



**Figure 4.** Display interface of Project Package workItem

- Project Package workflow configuration

Transitions						
	Draft	Planned	In Progress	Completed	Closed	On Hold
Draft	Set to Draft	Set to Plann			Set to Close	Set to On Hold
Planned			Set to In Pro		Set to Close	Set to On Hold
In Progress				Set to Comp	Set to Close	Set to On Hold
Completed			Set to In Pro		Set to Close	
Closed	Set to Draft		Resume wor			Set to On Hold
On Hold	Set to Draft	Set to Plann	Resume wor		Set to Close	

**Figure 5.** Workflow configuration overview for Project Package workItem

Figure 5 represents the entire workflow of the Project package which consists of six statuses:

1. **Draft:** The initial status of the Project Package
2. **Planned:** Once all Work Request are created and estimated
3. **In Progress:** When at least one of the Task have been started
4. **Completed:** Once all the tasks and work requests have been completed
5. **Closed:** Once the implementation has been verified and approved
6. **On Hold** (optional step): If for some reason, the work needs to be paused, this action will set all the descending items to On Hold status as well.

To help understanding the table in Figure 5, an analysis of the first row is given below:

Transitions						
Draft	Draft	Planned	In Progress	Completed	Closed	On Hold
Draft		Set to Plann-			Set to Close	Set to On Hk

**Figure 6.** Initial workflow transitions of Project Package workItem

In Figure 6, “Draft” status is the initial status, the row reads as follow: “When the project package is in “Draft” status, it is possible to make a transition to “Planned”, “Closed or “On Hold” status.” The table shall be read one row at the time, the leftmost column represents “from” and the header of each column represents “to”.

In addition to the status transition, some fields can be marked as required to reach certain status, in this example, to transition to “Planned” status, at least one linked Work Request to the Project Package is required.

- Project Package links configuration

A work package may be linked to a Work Request with the defined link role “Processes” so that a Work Request “processes” a Project Package.



**Figure 7.** Linked item creation shortcut menu from a Project Package item

Polarion allows to quickly create a linked workItem by clicking the “+” icon under the workItem title. Once a Work Request is linked and analyzed, all its data regarding cost and schedule will be automatically summed up and sent to the Project Package item.

## 6.2 Work Request

Work Requests workItems are assigned to a department. They represent several weeks of work that requires specific expertise corresponding to department core competence.

- Work Request Fields configuration

TP-2 +  
TP-4 - Design User Interface for upcoming ABCDXYZ product  
Created: 2021-02

Priority: <b>2 - Medium</b>	Assignee: <b>Mr Department Manager</b>	Cost Center ID: <b>ABCDXYZ</b>	Type: <b>Work Request</b>
Requested Due Date: <b>Before Q2 2021</b>	Status: <b>Draft</b>	Planned Start Date:	Project: <b>Thesis Project</b>
Due Date: <b>2021-02-19</b>	Resolution:	Planned End Date:	Author: <b>Mr Project Manager</b>
	Rough Estimate:		Submitted by:
	Started On:		

**Description**

The user interface must meet all the related requirements in attachments.

**Analysis**

**Rolled up estimates from children**

Initial Estimate:  
Time Spent:  
Remaining Estimate:  
Progress % (hrs):

**Comments**

Create Comment Collapse All Expand All View: **Tree**  Show resolved comments

**Figure 8.** Display interface of Work Request workItem

Work Requests contains additional fields, such as Analysis which must be filled in order to move forward from “Draft” to “Analyzed” status. The department-level team must analyze all the work required to complete this Work Request and break it down into tasks which are linked to the Work Request in a similar manner than Work Request are linked to Project Package.

- Work Request workflow configuration

Transitions							
	Draft	Analyzed	Planned	In Progress	On Hold	Closed	
Draft	-	Set to Analyz: -	-	-	Set to On Hi: -	Set to Close -	-
Analyzed	-	-	Set to Plann: -	-	Set to On Hi: -	Set to Close -	-
Planned	Set to Draft -	-	-	Set to In prc -	Set to On Hi: -	Set to Close -	-
In Progress	-	-	-	-	Set to On Hi: -	Set to Close -	-
On Hold	Set to Draft -	Set to Analyz: -	Set to Plann: -	Set to In Pro -	-	Set to Close -	-
Closed	Set to Draft -	-	-	Redo work ( -	Set to OnHo: -	Set to Close -	-

**Figure 9.** Workflow configuration for Work Request workItem

Figure 9 represents the workflow of the Work Request workItem which consists of six statuses:

1. **Draft:** The initial status of the item
2. **Analyzed:** Once the work amount has been evaluated and assigned
3. **Planned:** Once all the Task items are created, assigned and planned
4. **In Progress:** When at least one of the linked Task started
5. **Closed:** Once the implementation has been verified and approved
6. **On Hold (optional step):** If for some reason, the work needs to be paused, this action will set all the descending items to “On Hold” status as well.

Once the work analysis has been performed, the Work Request can move from “Analyzed” into “Planned” status, which means that all the tasks have been assigned to team members and are scheduled to meet the requested date.

### 6.3 Tasks

Tasks are the smallest work container and shall be executed within a couple of days. Figure 10 shows the interface of the Task workItem.

- Task fields configuration

Tasks contains crucial information, such as Due Date set by the team manager but also the planning information (Planned Start Date, Planned End Date) which is set by the team. The “Planned In” field offers the possibility to use Polarion Plans and add this task to the relevant plan (for example bi-weekly development sprints, in accordance with the team’s way of working).

TP-4 +  
TP-5 - Design initial styling for UI  
Created: 2021-02-01 01:3

Priority: 2 - Medium  
Due Date: 2021-02-26  
Planned Start Date: 2021-02-08  
Planned End Date: 2021-02-19

Assignee: Team Member A  
Planned In:  
Status: Draft  
Resolution:

Cost Center ID: ABCDXYZ  
Initial Estimate: 20  
Time Spent:  
Remaining Estimate: 20  
Progress % (hrs.):

Type: Task  
Author: Mr Team Manager

**Description**  
Design initial style for UI according to attached requirements

**Work Records**

**Comments**  
Create Comment | Collapse All | Expand All | View: Tree - | Show resolved comments

**Linked Work Items**

Suspect	Role	Title	Project	Revision	Status
Processes		TP-4 - Design User Interface for upcoming ABCDXYZ product	Thesis Project		

**Figure 10.** Display interface of Task workItem

“Work Records” field can be used to register all the worked hours, when a team member work on this specific task, he may record the time spent on this task each day, this way the remaining estimate and total time spent is updated and summed up in the parent Work Request which is carried up into the parent Project Package.

- Task workflow configuration

Workflow for tasks is rather simple and takes advantage of workflow scripts functionality. In accordance with Figure 11, when a task is set to “In Progress”, the status of the parent work request will be change to “In Progress” automatically and all estimates and time spent are automatically updated. Once the parent Work Request is in “In Progress” state, the parent Project Package is also set to “In Progress” automatically, this way, there is no need to manually inform the team manager that the work have started already, the Project Manager can immediately track the progress and the estimated completion date directly from the Project Package item.



Transitions				
	Draft	In Progress	On Hold	Closed
Draft		Set to In prc	Set to On Hc	Set to Close
In Progress	Set to Draft		Set to On Hc	Set to Close
On Hold	Set to Draft	Resume wor		Set to Close
Closed	Set to Draft			

**Figure 11.** Sample workflow configuration for Task workItem

- Task links configuration

There are only three layers of workItem types, so the tasks are only linked to their parent work request. Tasks do not have children's items.

In addition to the "Processes" link role, one may use "Depends On" role. For example, a task assigned to Team 1 depends on a task assigned to Team 2 so that Team 1 cannot complete their task unless Team 2 have finished implementation. This information is very useful and can be highlighted in dynamic reports.

## 7 POLARION ADVANCED SCRIPTING

This chapter focuses on Polarion scripting capabilities with two examples showing what can be done with workflow scripting and LiveReport scripting using the Highcharts library.

### 7.1 Workflow scripting

As explained in the Task workflow definition, it is possible to automate actions in Polarion when the status of an item is changed.

When a team member starts executing a Task, the status is set from “Draft” to “In Progress” and the parent Work Request is automatically set to “In Progress” status as well, this ensures the Department manager is notified that the work started.

In order to automatically propagate the change of status, a workflow script is required and detailed below.

```
// Fetch the workItem where the status is being changed
var workitem = workflowContext.getTarget();
// Returns the list of all linked workitems

var linkedWi = workitem.getLinkedWorkItemsStructsDirect();
// Check that the user have permission to edit the item

if (workitem.can().modify()) {
// Foreach linked workitem
  for (i = 0; i < linkedWi.size(); i++) {
// Returns the ID of the link role

    linkRoleResult =linkedWi[i].getLinkRole().getId();
// If link role match the desired link role
    if(linkRoleResult === linkRole){
// Return the parent workItem
      var parent= linkedWi[i].getLinkedItem();
// Check permission to edit the parent item
      if (parent.can().modify() == true) {
```

```

// Returns all available status transition of the parent
    var actions = parent.getAvailableActions();
// Make sure that the parent is of the correct type (parent of task must be
Work Request)
    if(parent.getType().getId() === parentType) {
        for(j=0; j < actions.length; j++) {
// Select the desired action (eg: Set to In Progress)
            if (actions[j].getNativeActionId() == inProgressActionId) {
// Perform the status change on the parent

                parent.performAction(actions[j].getActionId());
// Save the parent item
                parent.save();
}}}}}}

```

**Code sample 1.** Workflow script to modify linked workItems status

This script is loaded into the task’s workflow “Start Progress” action configuration where the script name and parameters are given.

The workflow script configuration must include two main parameters as well as three custom parameters:

- **Apptasks:** Is a mandatory parameter to tell Polarion which engine shall be used; in this case it is set to “js” for JavaScript.
- **Script:** Is a mandatory parameter to indicate the script’s filename on the server.
- **linkRole:** Is the name of the link role to track, in this case “processes”.
- **inProgressAction:** Is the targeted status name for “In Progress” status.
- **parentType:** Is the workItem type of the parent item (in case of Task, the parent item should be of type Work Request)

**Parameter for: ScriptFunction**

Close

**Parameters**

Name	Value	Actions
Apptasks	js	-
script	changeParentStatus.js	-
linkRole	processes	-
inProgressAction	startProgress	-
parentType	workRequest	-
<input type="text"/>	<input type="text"/>	+

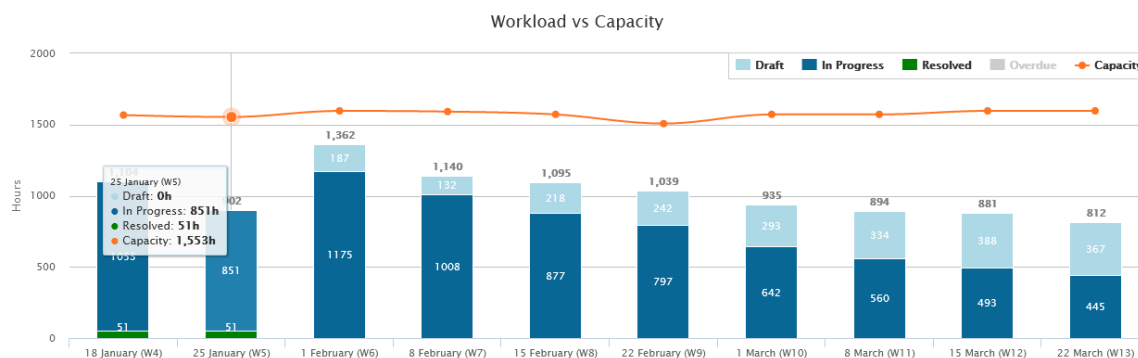
**Figure 12.** Workflow configuration for "Progress" action in Task workflow

## 7.2 Highcharts Live Report Development

The Highcharts library is very flexible and offers a lot of customization options, one may find more information from the Highcharts API documentation. /5/

The following chart consists of a bar chart calculating past and future workload of selected department and a line representing available work hours capacity. The purpose of this chart is to provide visibility of the department resource allocation. The workload shall always be below the capacity to avoid over-booking the resources. This report can be displayed for each department to compare resource allocation between them.

## Workload vs. Capacity (Tasks)



**Figure 13** Workload vs Capacity Highcharts chart built from Polarion data

The data related to workload comes from Task workItems in Polarion, the data regarding capacity comes from the user calendar feature in Polarion where one can set their regular working hours as well as exceptions, such as vacations or national holidays. The elaboration of such chart can be divided into two steps, the collection of data from Polarion database and the chart configuration using Highcharts API.

### 7.2.1 Retrieving Data from Polarion Database

The first step of retrieving data from Polarion consists of building a Lucene query. In this example, there are only three query parameters: “project.id” which limits the item search to the current project only, “type” which limits the search to workItems of type “Task” only and “resolvedOn” date filter which excludes all tasks that have been closed before the beginning of the chart’s timeline (two weeks before today). The language used in these live report pages is Apache Velocity 1.4.

```
#set($query = "project.id:demoProject AND type:task AND NOT
resolvedOn:[10000000 TO $today-14d$]")
#set($items = $transaction.workItems().search().query("$query"))
```

**Code sample 2.** Query Polarion workItems using Apache Lucene query

The first line defines the *\$query* variable of type String containing the Lucene query. The second line retrieve the corresponding workItems into the *\$items* variable which is an array of workItem objects.

Once the data is extracted, the data is parsed for each period in the chart. The work records are distributed to their corresponding time stamps and the remaining estimated hours of the tasks is proportionally distributed between “Planned Start Date” until the “Planned End Date”. If they are not “In Progress” status, in that case, the distribution of the remaining working hours is done between “Today” and “Planned End Date”. After which, the user calendar information is fetched.

```
#foreach($user in $teamMembers)
    ## Fetch the calendar object of that user
    #set($workingCalendar =
    $planningManager.getWorkingCalendarForUser($user.id,true))

    ## Count from 1 to x (amount of days in this preiod/bar/step)

    #foreach($day in [1..$daysInPeriod
        #set($dailyHours = 0) ## How many hours in that day
    ## Fetch specific day informaiton of working calendar of the current user
    #set($workingDay = $workingCalendar.getWorkingDay($dailyLoop.getTime,true)
    ## If this day is a working day
    #if($workingDay.isWorking)
        #foreach($entry in $workingDay.getWorkingTimes)
            #set($dailyHours =
            $math.div($math.sub($entry.getTime,$entry.getFromTime), 3600000))
        ## Add daily hours (ex. 8h) to the daily capacity of that person.
        #set($capacityUserStep = $math.add($capacityUserStep,$dailyHours))
        #end
    #end
    $!dailyLoop.add(6,1) ## Add one day
    #end
## Add the capacity of user in the period to the overall capacity of that step
for all users

    #set($capacityStep = $math.add($capacityStep,$capacityUserStep))
#end
```

**Code sample 3.** Velocity code to retrieve data from a user's calendar

At this point, workItems have been saved in an array and the hours estimate are saved in dedicated variables. The capacity of all involved users is also stored in *\$capacityStep* variable.

### 7.2.2 Highcharts Options

Highcharts based charts requires configuration, it is more convenient to switch to JavaScript language within the same script to define JSON-like parameter according to the Highcharts API. /5/

All parameters are stored in the *options* JavaScript object.

```
var options = {
  title: {
    text: 'Workload vs Capacity',
  },
  xAxis: {
    categories: $periodsArray,
    crosshair: true,
  },
  yAxis: { // First axis, workload
    min: 0,
    title: {
      text: 'Hours',
    },
  },
  plotOptions: {
    column: {
      stacking: 'normal',
    },
    series: {
      cursor: 'pointer',
      point: {
        events: {
          click: function() {window.open(this.options.url);}
        }
      }
    }
  }
}
```

**Code sample 4.** Highcharts parameters in JSON format

The series data which consists of processed data extracted from Polarion is then passed to Highcharts for each series, this data contains the hours value for each period as well as URL redirecting to the list of selected tasks.

```

series: [
  {
    name: 'Draft',
    type:'column',
    color: 'lightblue',
    data: $draft,
  },
  {
    name: 'In Progress',
    type:'column',
    color: 'rgb(8,103,149)',
    data: $inProgress,
  },
  {
    name: 'Resolved',
    type:'column',
    color: 'green',
    data: $closed,
  },{
    name: 'Capacity',
    type: 'spline',
    color: 'rgb(255,115,33)',
    data: $capacity,
  }
]

```

**Code sample 5.** Highcharts series options in JSON format

Formulas for calculating the average hours depending on each status and available date field data are not detailed in this publication but it is important to note that the process remains the same regardless of the type of chart needed and consists of four steps:

1. Extracting data from Polarion workItems through Lucene query.
2. Parsing the relevant data into series arrays.
3. Configuring Highcharts options and passing the data series array.
4. Building the Highcharts chart from the options.



**Note:** The same chart can be rendered for Work Requests or Project Package simply by changing the “type” filter in the main query.

This results in a dynamic live chart where each column can be clicked to open the list of included tasks. This scalable chart is used by Project Managers, Department Managers and Team Managers and offers a quick overview of resource availability and upcoming workload with the possibility to see the details of the items.

## 8 CONCLUSIONS

This thesis work highlights the complexity of project management and coordination challenges within a large enterprise and proposes a generic solution to solve this problem with the help of Polarion software.

Changing work methodology or tools is always complex when considering many people and factors. Such a change has a significant impact on work habits of employees and may lead to additional challenges, such as change resistance or frustration when the tool complexity increases. Consulting as much persons as possible is a good way to start such a project.

Polarion is a powerful and flexible tool that requires a large investment and effort to fit in a complex enterprise environment but will provide great added value to any given productive.

Overall, the task management project was a success, it strengthened communication across all departments involved and provided an interface for management to oversee ongoing activities within the company.

In this project, human factor or other challenges that may be encountered were not taken into account. Polarion is also a great solution for requirement management which is also part of project management. Requirements management is a discipline of its own, the process of setting up requirement management configuration using Polarion have similarities with setting up of the task management project. At first, the roles and responsibilities are defined, then comes Polarion workItems configuration followed by workflow and fields definition, the final setup can be continuously improved later with custom reports and automation scripts.

## REFERENCES

- /1/ Polarion - software. Accessed 8.4.2021.  
<https://polarion.plm.automation.siemens.com>
- /2/ Highcharts JS API Reference. Accessed 8.4.2021.  
<https://api.highcharts.com/highcharts/>
- /3/ Polarion OpenAPI Transaction documentation. Accessed 8.4.2021.  
<https://almdemo.polarion.com/polarion/sdk/doc/javadoc/com/polarion/alm/tracker/ITrackerService.html>
- /4/ Polarion Rendering API documentation. Accessed 8.4.2021.  
<https://almdemo.polarion.com/polarion/sdk/doc/javadoc-rendering/com/polarion/alm/shared/api/transaction/ReadOnlyTransaction.html>
- /5/ Highcharts Library documentation. Accessed 8.4.2021.  
<https://www.highcharts.com/docs/index>
- /6/ Apache Lucene - Query Parser Syntax. Accessed 8.4.2021.  
[https://lucene.apache.org/core/2\\_9\\_4/queryparsersyntax.html](https://lucene.apache.org/core/2_9_4/queryparsersyntax.html)