Process Management for ICT Engineering

Assignment 3: Process Management Documentation

Jordi Rafael Lazo Florensa

28th April 2022

Software Technology Engineering

Teacher: Poul Væggemose

IT-PME1X-S22

VIA University
College

Contents

List of Tables

List of Figures

1 Abstract

Most software projects fail. In fact, the Standish group reports that over 80% of projects are unsuccessful either because they are over budget, late, missing function, or a combination. Moreover, 30% of software projects are so poorly executed that they are canceled before completion. To avoid these problems it is recommended make use of the best practice models and standards. It is essential to understand and apply the different improvement process models such as CMMI, TMMI and others. For this reason, this document aims to explain, understand and comprehend what these process evaluation and improvement programs consist of and how they can be applied in engineering projects.

2 Introduction

Process management is the organizational discipline that provides tools and resources for analyzing, defining, optimizing, monitoring, and controlling business processes and for measuring and driving improved performance of interdependent business processes. Therefore, during the Process Management for ICT Engineering course, professor Poul Væggemose has defined and explained different process evaluation programs all this with the intention of remark the importance of making use of the best practice models and standards such as CMMI, IEEE, ISO, TMMI and much more. These standards are necessary to ensure quality in projects, improve projects performance and handle change management in projects. Also these systems optimize and accelerate processes, increasing efficiency. By ensuring that work is handled in a consistent manner and that tasks are delivered to the right users at the right time. Last but not least, this document will explain the different methods, guidelines and strategies for setting goals and objectives as well as testing concepts. Finally, the conclusions of the other assignments presented during the course will be presented.

3 CMMI model levels 1-5

Capability Maturity Model Integration (CMMI) is a process improvement approach that provides organizations with the essential elements of effective processes that ultimately improve their performance.

The benefits associated with CMMI models are: Improvement in productivity, quality and increase in cycle time thus improving the customer satisfaction, meeting business objectives, improvement in business and growth.

The purpose of the CMMI model is to assess the maturity of an organization's processes and to provide guidance on improving processes, with a goal of improved products. Also, CMMI is a model for risk management and provide a way to measure an organization's ability to manage risk. The ability to manage risk factors factors into an organizations ability to deliver high-quality products. Another perspective on managing risk is how well an organization will perform under stress. A high maturity, high capability organization can easily respond to unexpected, stressful events. A low maturity and lower capability organization tends to panic under stress, blindly follow obviated procedures, or throw out all process altogether and retrench back to chaos.

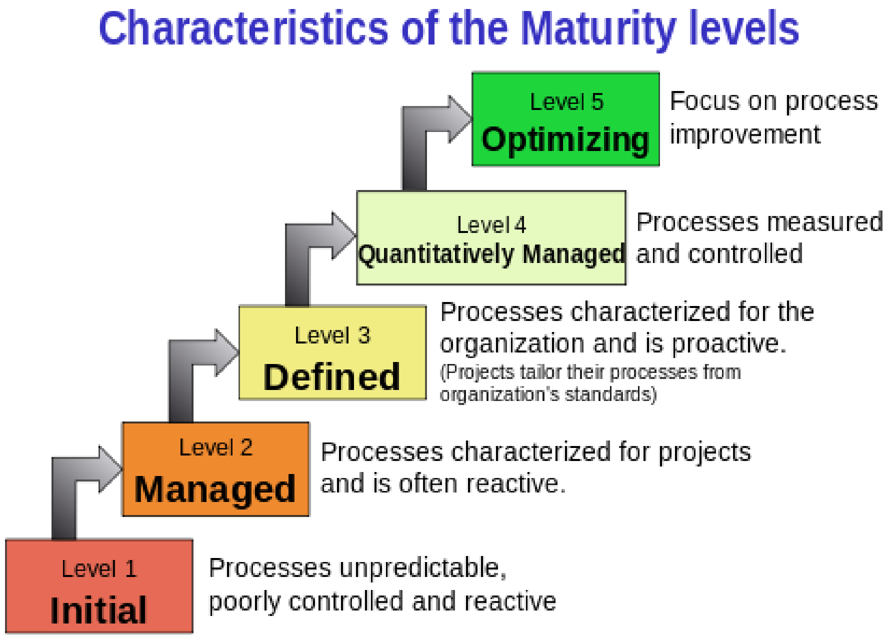
Characteristics of the Maturity levels
Level 5
Optimizing
Focus on process improvement
Level 4
Quantitatively Managed
Processes measured and controlled
Level 3
Defined
Processes characterized for the organization and is proactive.
(Projects tailor their processes from organization's standards)
Level 2
Managed
Processes characterized for projects and is often reactive.
Level 1
Initial
Processes unpredictable, poorly controlled and reactive

Figure 1: *Characteristics of the CMMI levels.*

3.1 Level 1: Initial

Processes are seen as unpredictable, poorly controlled, and reactive. Businesses in this stage have an unpredictable environment that leads to increased risks and inefficiency. Success in these organizations depends on the competence and heroics of the people in the organization and not on the use of proven processes.

- Organizations often produce products and services that work; however, they frequently exceed the budget and schedule of their projects.
- Organizations are characterized by a tendency to over commit, abandon processes in the time of crisis, and not be able to repeat their past successes.

3.2 Level 2: Managed

Processes are characterized by projects and are frequently reactive.

At maturity level 2, an organization has achieved all the specific and generic goals of the maturity level 2 process areas. In other words, the projects of the organization have ensured that requirements are managed and that processes are planned, performed, measured, and controlled.

The process discipline reflected by maturity level 2 helps to ensure that existing practices are retained during times of stress. When these practices are in place, projects are performed and managed according to their documented plans.

At maturity level 2, requirements, processes, work products, and services are managed. The status of the work products and the delivery of services are visible to management at defined points.

Commitments are established among relevant stakeholders and are revised as needed. Work products are reviewed with stakeholders and are controlled.

The work products and services satisfy their specified requirements, standards, and objectives.

3.3 Level 3: Defined

At this level, a business can accomplish goals set forward in Levels 2 and 3. Their processes are well-documented and understood. Companies who fall into this category can explain what each of their processes do. They are also well-aware of the scope of those processes and the standards they conform to. The procedures are proactive in trying to ensure that the business is continually improving in its efforts.

3.4 Level 4: Quantitatively Managed

At this level, the business is managed using statistical data and techniques that utilize the information to improve them in a manner that affects those metrics. Companies at Level 4 achieve the goals outlined in the previous three levels as well as this one. At Level 3, only the quality of the improvements can be predicted accurately. At Level 4, the exact value of the development can be determined precisely.

3.5 Level 5: Optimizing

This maturity level focuses on constant improvement. By focusing on the quantitative methodology established at the previous level, a business can innovate to optimize their current processes. This innovation is the core of the improvement process at maturity level 5. Feedback from the previous standard can help the business be aware of whether it's hitting its goals or if something needs to change. Companies at Level 5 also can achieve all the goals outlined in the previous levels.

4 SCAMPI

The Standard CMMI Appraisal Method for Process Improvement (SCAMPI) is the official Software Engineering Institute (SEI) method to provide benchmark-quality ratings relative to Capability Maturity Model Integration (CMMI) models. SCAMPI appraisals are used to identify strengths and weaknesses of current processes, reveal development/acquisition risks, and determine capability and maturity level ratings. They are mostly used either as part of a process improvement program or for rating prospective suppliers. The method defines the appraisal process as consisting of preparation; on-site activities; preliminary observations, findings, and ratings; final reporting; and follow-on activities. It is applicable to a wide range of appraisal usage modes, including both internal process improvement and external capability determinations.

4.1 SCAMPI Class A

A SCAMPI Class A appraisal, the most rigorous of the Software Engineering Institute (SEI) process appraisal methods, is typically conducted when an organisation has implemented a planned set of process improvements and needs to formally benchmark its process relative to the CMMI.

A SCAMPI A is the only appraisal method that provides a recognized CMMI maturity level rating.

A SCAMPI A appraisal is typically conducted when an organisation has implemented a number of significant process improvements and needs to formally benchmark its process relative to the CMMI. A SCAMPI A is the only appraisal method that provides CMMI maturity level or Capability Level ratings.

Class A methods must satisfy all of the Appraisal Requirements for CMMI (ARC) requirements, and at the present time are the only methods considered suitable for providing ratings for benchmarking. Developers of Class A methods also have the option of supporting the conduct of 15504-conformant appraisals. An example of a Class A method is the Standard CMMI Appraisal Method for Process Improvement (SCAMPI).

4.2 SCAMPI Class B

A SCAMPI Class B appraisal, one of three SEI appraisal methods, helps an organisation understand, with a relatively high degree of confidence, the status of its software and systems engineering process relative to the CMMI. A SCAMPI B is often performed when an organisation needs to accurately assess its progress towards a target CMMI maturity level.

A SCAMPI B is called for when an organisation needs to assess its progress towards a target CMMI maturity level, but at a lower costs than a SCAMPI A. SCAMPI B appraisals provide detailed findings and indicate the likelihood that the evaluated practices would be rated as satisfactorily implemented in a SCAMPI A appraisal.

4.3 SCAMPI Class C

SCAMPI Class C appraisals, the least formal of the SEI's suite of appraisal methods, are highly flexible and can be conducted to address a variety of needs. Typically much shorter in duration than SCAMPI A and SCAMPI B.

SCAMPI C appraisals are often performed for reasons such as:

- A quick gap analysis of an organisation's process relative to the CMMI.
- The adequacy of a new process before it is implemented
- The implementation of a process
- An organisation's readiness for a SCAMPI A
- The selection of a supplier.

SCAMPI C appraisals are shorter and more flexible than SCAMPI A and SCAMPI B and are conducted to address a variety of special needs, from a quick gap analysis to determining an organisation's readiness for a SCAMPI A.

5 Lewin's change model 3 phases

Lewin's Change Management Model is a comprehensive change model aiming to understand why change occurs and what must be done to deliver change in the most seamless way possible. Lewin developed the change model as a way to illustrate how people react when facing changes in their lives.

The three stages of this process include unfreezing (the person has an existing state), moving or changing towards new ways of being, and then refreezing into a new state altogether.

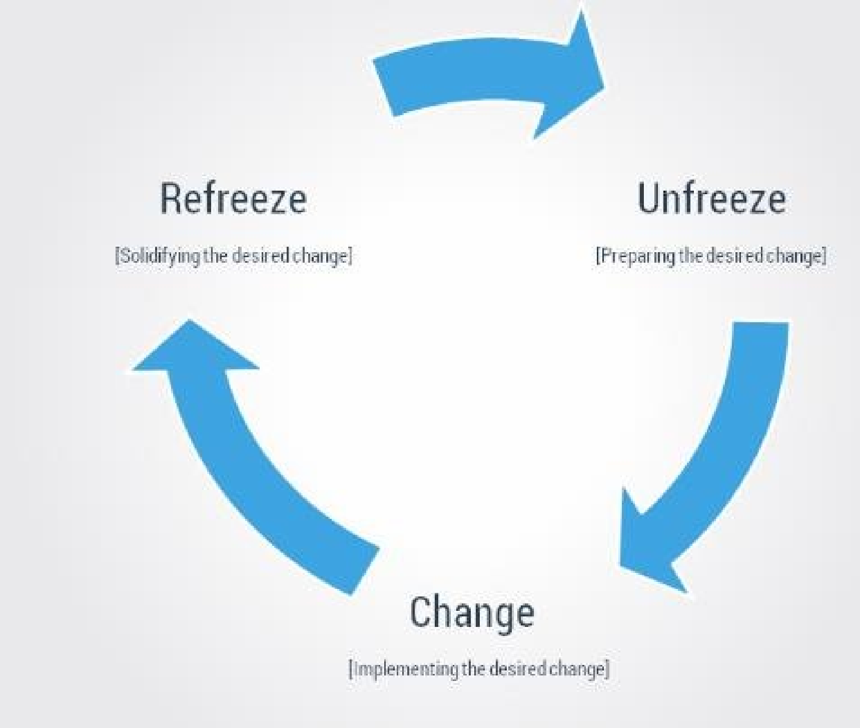
Refreeze
[Solidifying the desired change]
Unfreeze
[Preparing the desired change]
Change
[Implementing the desired change]

Figure 2: *Lewin's 3 stage model.*

5.1 Phase 1: Unfreeze

This first stage of change involves preparing the organization to accept that change is necessary, which involves breaking down the existing status quo before you can build up a new way of operating.

Key to this is developing a compelling message showing why the existing way of doing things cannot continue.

This first part of the change process is usually the most difficult and stressful.

5.2 Phase 2: Change

After the uncertainty created in the unfreeze stage, the change stage is where people begin to resolve their uncertainty and look for new ways to do things. People start to believe and act in ways that support the new direction.

The transition from unfreeze to change does not happen overnight: people take time to embrace the new direction and participate proactively in the change.

In order to accept the change and contribute to making it successful, people need to understand how it will benefit them. Not everyone will fall in line just because the change is necessary and will benefit the company. This is a common assumption and a pitfall that should be avoided.

Time and communication are the two keys to the changes occurring successfully. People need time to understand the changes, and they also need to feel highly connected to the organization throughout the transition period.

5.3 Phase 3: Refreeze

When the changes are taking shape and people have embraced the new ways of working, the organization is ready to refreeze. The outward signs of the refreeze are a stable organization chart, consistent job descriptions, and so on.

The refreeze stage also needs to help people and the organization to internalize or institutionalize the changes. This means making sure that the changes are used all the time, and that they are incorporated into everyday business. With a new sense of stability, employees feel confident and comfortable with the new ways of working.

6 Systematic - CMMI level 5

Systematic A/S is an international software company founded in Denmark in 1985. Systematic has more than 1100 employees and provides software solutions to private and public sectors in more than 50 countries.

It focuses on six core business areas: Digitalisation, Healthcare, Defence, Intelligence National Security, Library Learning og Renewables Utilities.

The users primarily comprise nurses, soldiers, librarians, doctors, the police, teachers, home care workers, offshore coordinators and administrative employees in the public sector.

Systematic is one of the best companies worldwide at delivering software on time. In 2021, 99% of all Systematic's projects were delivered before or on time, which is quite unique for the IT industry.

Since 2005, the company has been CMMI-appraised at Level 5; this is the highest possible level, and demonstrates that they are among the best worldwide at delivering software solutions on time, within budget and to the agreed quality.

The CMMI Level 5 appraisal demonstrates that the company is a mature organisation with the discipline required to manage, predict and optimise the development processes, while carrying out ongoing follow-ups and adjustments to ensure that they always deliver the best results to their customers.

To have the CMMI level 5 means that customers of Systematic have advantage of:

- They receive high-quality solutions at competitive prices.

- The projects are delivered on time and within budget.
- They will experience accurate project planning and faster time-to-market for our solutions.
- They receive a well-designed and easy-to-maintain solution at a competitive price.

Figure 3: *Systematic CMMI level 5 certificate.*

7 TMMI model levels 1-5

The Testing Maturity Model Integration (TMMI) was based on the Capability Maturity Model Integration, and first produced by the Illinois Institute of Technology.

Its aim to be used in a similar way to CMMI, that is to provide a framework for assessing the maturity of the test processes in an organisation, and so providing targets on improving maturity.

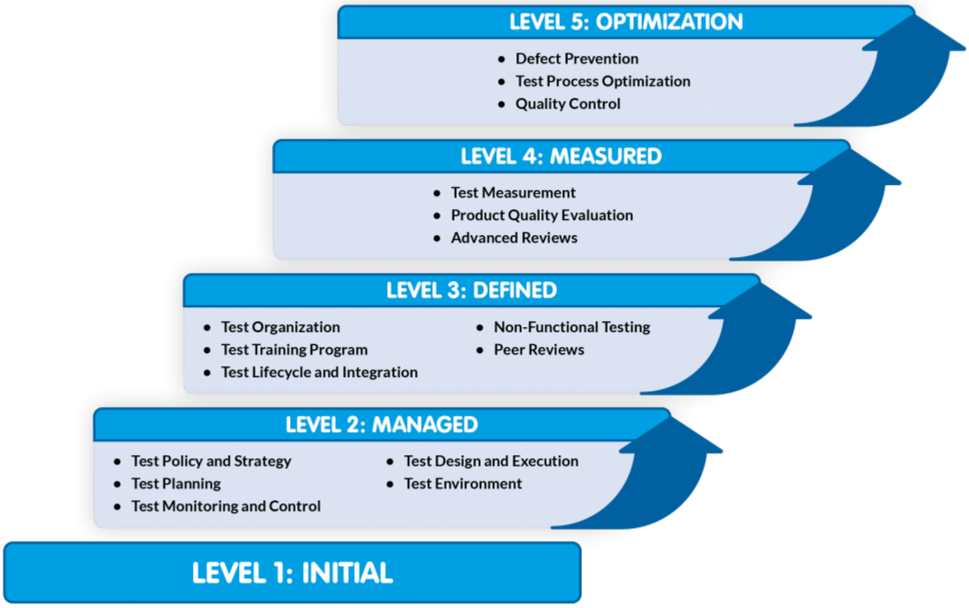

Figure 4: *Characteristics of the TMMI levels..*

7.1 Level 1: Initial

At this level an organisation is using ad hoc methods for testing, so results are not repeatable and there is no quality standard.

7.2 Level 2: Managed

Where the fundamental test approach is established and managed, which may vary from project to project within an organization with test policy and strategy being a focus area. This provides direction for planning monitoring and control, test design techniques with each project having control over test execution. At level 2 test environments are also in place for projects.

7.3 Level 3: Defined

Establishes that all projects are following the same standards and procedures throughout the organizations or organizational unit. Level 2 is still being done, and teams are now organized, test training programs exist, test is integrated into the development life cycle and integrated into all projects from early in development. Non-functional testing is planned and executed in all projects and reviews are used in each project as well.

7.4 Level 4: Measured

Measurement of activities and outcomes are thoroughly applied early in all projects to establish at each stage of development the projects are as defect free as possible. Advanced reviews are in use throughout all projects building on the review practices introduced at 3.

7.5 Level 5: Optimization

At this level all activities and outcomes are assessed and optimize activities are in place to ensure continuous improvement toward defect prevention and optimized quality.

8 Reflection on CMMI, TMMI and Lewin's models

The main difference between CMMI and TMMI is that the CMMI framework primarily focuses on all the practices of software development. On the other hand, the TMMI framework mainly focuses on all the processes that may eventually lead to the betterment of quality enhancement of the software.

In addition, other outstanding differences are:

- CMMI is all about defect management whereas TMMI is all about defect prevention
- CMMI is focuses on execution of the softwares whereas TMMI focuses on monitoring of the softwares.
- CMMI has both staged and continuous processes whereas TMMI has only staged processes.
- CMMI is primarily focused on software improvement or upgrading whereas TMMI is focused on the debugging the goals and it's policies.
- CMMI is a standalone process whereas TMMI has to work completely following CMMI.

CMMI can help companies improve the quality of their products along with the production services by providing them with everything they need to do so. In addition to a process model, CMMI is also a behavioral-based model. CMMI is all about defect management whereas TMMI is all about defect prevention. Through TMMI or Test Maturity Model Integration, organizations under the IT sector can improve their software testing practices and improve the company's IT standards.

Regarding to Lewin's models change processes often fail because an organization does not succeed in communicating the need for change to their employees. People/ employees will only be able to make concessions and let go of the old situation when they are aware of the need for change.

The only way in which management will succeed in convincing staff of this need, is openness, transparency and honesty. When the employees are involved in the process, they will understand the need for change and they can work on their awareness of the change.

It is an excellent method to bring about a mentality change among employees and creating awareness of the advantages of change.

Lewin's model recognizes the importance of the interdependence of units and departments within an organization. Yet it assumes that organizations act in a static manner, when in reality the workplace is a dynamic and turbulent place.

As a result, the author of the model receives criticism that the model is too simple and mechanistic. It is therefore not applicable for an entire organization in radical and disruptive change. It is, however, useful for incremental change.

9 Testing concepts

Concept testing is defined as a research method that involves asking customers questions about your concepts and ideas for a product or service before actually launching it.

Software testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

Some prefer saying Software testing definition as a White Box and Black Box Testing. In simple terms, Software Testing means the Verification of Application Under Test (AUT). This Software Testing course introduces testing software to the audience and justifies the importance of software testing.

10 Other project management concepts

Project management is the process of leading the work of a team to achieve all project goals within the given constraints. This information is usually described in project documentation, created at the beginning of the development process. The primary constraints are scope, time, and budget. The secondary challenge is to optimize the allocation of necessary inputs and apply them to meet pre-defined objectives.

The core components of project management are:

- Defining the reason why a project is necessary.
- Capturing project requirements, specifying quality of the deliverables, estimating resources and timescales.
- Preparing a business case to justify the investment.
- Securing corporate agreement and funding.
- Developing and implementing a management plan for the project.
- Leading and motivating the project delivery team.
- Managing the risks, issues and changes on the project.
- Monitoring progress against plan.
- Managing the project budget.
- Maintaining communications with stakeholders and the project organisation.
- Provider management.
- Closing the project in a controlled fashion when appropriate.

11 SMART principle

S.M.A.R.T. is a mnemonic acronym, giving criteria to guide in the setting of goals and objectives, for example in project management, employee-performance management and personal development.

The SMART principle defends that the objectives should be:

- Specific: state exactly what will need to be done.
- Measurable: be clear what success will look like.
- Agreed: make sure others share it, eg coach, team.
- Realistic: know it is practical – steps can be taken to do it.
- Time-phased: state when it will be achieved.

Example of SMART goal setting:

	Person A
S	I will increase how much exercise I do
M	I will do an average of 60 minutes of moderate intensity activity a day
A	I'm going to do it with a friend
R	I can do it by walking daily and going to the gym twice a week
T	I will achieve it by the end of this summer term

Table 1: *Example of SMART principle.*

12 Reflection on SMART principle

Often, individuals or businesses will set themselves up for failure by setting general and unrealistic goals such as "I want to be the best at X." This goal is vague, with no sense of direction.

SMART goals set up for success by making goals specific, measurable, achievable, realistic, and timely. The SMART method helps push further, gives a sense of direction, and helps organize and reach goals.

In addition, is an effective tool that provides the clarity, focus and motivation to achieve personal and business goals. SMART goals are also easy to use by anyone, anywhere, without the need for specialist tools or training.

Some people believe that SMART doesn't work well for long-term goals because it lacks flexibility, while others suggest that it might stifle creativity.

13 V-Model, SCRUM, Kanban and UP process

13.1 V-Model

In software development, the V-model represents a development process that may be considered an extension of the waterfall model, and is an example of the more general V-model. Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical V shape.

The various phases of the V-model are as follows:

Design Phase:

- Requirement Analysis: This phase contains detailed communication with the customer to understand their requirements and expectations. This stage is known as Requirement Gathering.
- System Design: This phase contains the system design and the complete hardware and communication setup for developing product.
- Architectural Design: System design is broken down further into modules taking up different functionalities. The data transfer and communication between the internal

modules and with the outside world (other systems) is clearly understood.

- Module Design: In this phase the system breaks down into small modules. The detailed design of modules is specified, also known as Low-Level Design (LLD).

Testing Phases:

- Unit Testing: Unit Test Plans are developed during module design phase. These Unit Test Plans are executed to eliminate bugs at code or unit level.
- Integration testing: After completion of unit testing Integration testing is performed. In integration testing, the modules are integrated and the system is tested. Integration testing is performed on the Architecture design phase. This test verifies the communication of modules among themselves.
- System Testing: System testing test the complete application with its functionality, inter dependency, and communication.It tests the functional and non-functional requirements of the developed application.
- User Acceptance Testing (UAT): UAT is performed in a user environment that resembles the production environment. UAT verifies that the delivered system meets user's requirement and system is ready for use in real world.

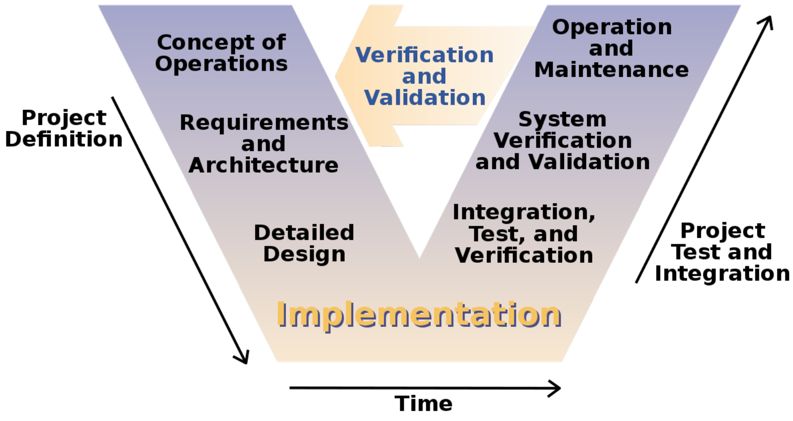
Concept of Operations
Verification and Validation
Operation and Maintenance
Project Definition
Requirements and Architecture
System Verification and Validation
Detailed Design
Integration, Test, and Verification
Project Test and Integration
Implementation
Time

Figure 5: *The V-Model structure.*

13.2 SCRUM

Within project management, scrum, is a framework for developing, delivering, and sustaining products in a complex environment.

It is designed for teams of ten or fewer members, who break their work into goals that can be completed within time-boxed iterations, called sprints, no longer than one month and most commonly two weeks. The scrum team assess progress in time-boxed daily meetings of 15 minutes or less, called daily scrums (a form of stand-up meeting). At the end of the sprint, the team holds two further meetings: the sprint review which demonstrates the work done to stakeholders to elicit feedback, and sprint retrospective which enables the team to reflect and improve.

- Scrum Team

  - Developers: the developers carry out all work required to build increments of value every sprint (include researchers, architects, designers, data specialists, statisticians, analysts, engineers, programmers, and testers, among others).
  - Product Owner: the product owner, representing the product's stakeholders and the voice of the customer, is responsible for delivering good business results.
  - Scrum Master: is accountable for removing impediments to the ability of the team to deliver the product goals and deliverables. It ensures that the scrum framework is followed by coaching the team in scrum theory and concepts, often facilitating key sessions, and encourages the team to grow and to improve.

- Scrum Events
  - The Sprint: is the basic unit of development in scrum. The sprint is a time-boxed effort, that is, the length is agreed and fixed in advance for each sprint and is normally between one week and one month.
  - Sprint Planning: sprint planning is an event in scrum that kicks off the sprint. The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved. Sprint planning is done in collaboration with the whole scrum team.
  - Daily Scrum: the purpose of the Daily Scrum is to inspect progress toward the Sprint Goal and adapt the Sprint Backlog as necessary, adjusting the upcoming planned work.
  - Sprint Review: the purpose of the Sprint Review is to inspect the outcome of the Sprint and determine future adaptations.
  - Sprint Retrospective: is a recurring meeting held at the end of a sprint used to discuss what went well during the previous sprint cycle and what can be improved for the next sprint.

- Scrum Artifacts
  - Product Backlog: is a breakdown of work to be done and contains an ordered list of product requirements that the team maintains for a product.
  - Sprint Backlog: is the subset of items from the product backlog intended for developers to address in the upcoming sprint.
  - Increment: an Increment is the latest stable version of their product that is usable by the users.

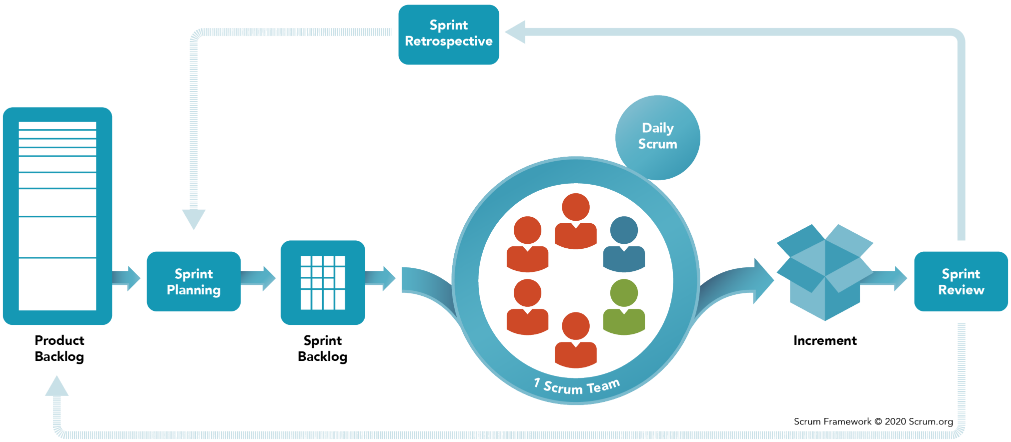

Figure 6: *Scrum framework.*

13.3 Kanban

"Kanban" is the Japanese word for "visual signal".

It is a popular lean workflow management method for defining, managing, and improving services that deliver knowledge work. It helps to visualize work, maximize efficiency, and improve continuously.

Work is represented on Kanban boards, allowing to optimize work delivery across multiple teams and handle even the most complex projects in a single environment.

Kanban boards use cards, columns, and continuous improvement to help technology and service teams commit to the right amount of work, and get it done.

- Elements of a kanban board:
  - Visual Signals: Kanban teams write all of their projects and work items onto cards, usually one per card. For agile teams, each card could encapsulate one user story. Once on the board, these visual signals help teammates and stakeholders quickly understand what the team is working on.
  - Columns: Each column represents a specific activity that together compose a "workflow". Cards flow through the workflow until completion. Workflows can be as simple as "To Do," "In Progress," "Complete," or much more complex.
  - Work In Progress (WIP) Limits: WIP limits are the maximum number of cards that can be in one column at any given time.
  - Commitment point: The commitment point is the moment when an idea is picked up by the team and work starts on the project.
  - Delivery point: is the end of a kanban team's workflow. The delivery point is when the product or service is in the hands of the customer.

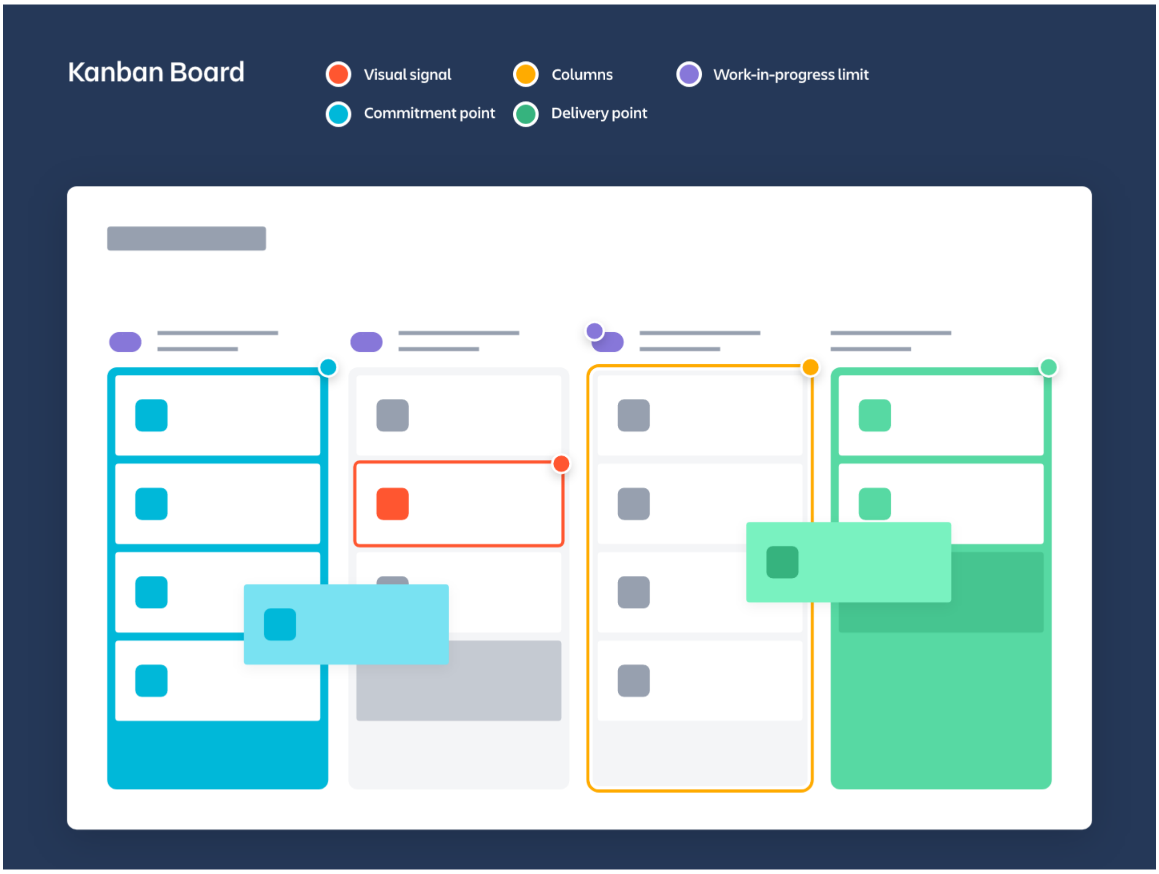

Figure 7: *Elements of a kanban board by Atlassian.*

13.4 UP Process

The Unified Software Development Process or Unified Process is an iterative and incremental software development process framework.

The system is developed incrementally over time, iteration by iteration, and thus this approach is also known as iterative and incremental software development. The iterations are spread over four phases where each phase consists of one or more iterations:

This process divides the development process into five phases:

- Inception: It is used to prepare basis for the project, including preparation of business case, establishing project scope and setting boundaries, outlining key requirements, and possible architecture solution together with design tradeoffs, identifying risks, and development of initial project plan—schedule with main milestones and cost estimates.
- Elaboration: during this phase the project team is expected to capture a majority of system's requirements (e.g., in the form of use cases), to perform identified risk analysis and make a plan of risk management to reduce or eliminate their impact on final schedule and product, to establish design and architecture (e.g., using basic class diagrams, package diagrams, and deployment diagrams), to create a plan (schedule, cost estimates, and achievable milestones) for the next (construction) phase.

- Construction: During this phase, the design of the system is finalized and refined and the system is built using the basis created during elaboration phase. The construction phase is divided into multiple iterations, for each iteration to result in an executable release of the system.
- Transition: the final project phase which delivers the new system to its end-users.

14 SWOT principle

SWOT analysis (or SWOT matrix) is a strategic planning and strategic management technique used to help a person or organization identify strengths, weaknesses, opportunities, and threats related to business competition or project planning.

The name is an acronym for the four components the technique examines:

- Strengths: strengths are things that an organization does particularly well, or in a way that distinguishes from others competitors. Think about the advantages of an organization has over other organizations. These might be the motivation of the staff, access to certain materials, or a strong set of manufacturing processes.
- Weaknesses: they are areas where the business needs to improve to remain competitive: a weak brand, higher-than-average turnover, high levels of debt, an inadequate supply chain, or lack of capital.
- Opportunities: elements in the environment that the business or project could exploit to its advantage.
- Threats: threats refer to factors that have the potential to harm an organization, such as supply-chain problems, shifts in market requirements, or a shortage of recruits.

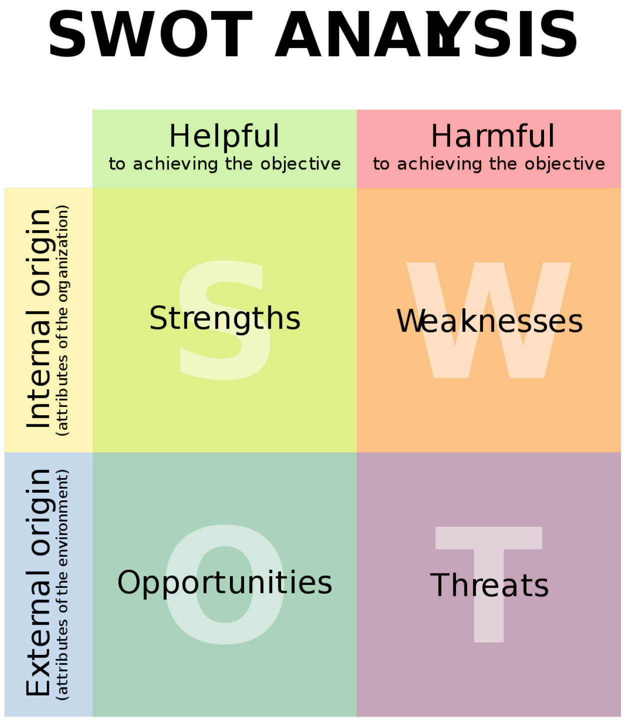
SWOT ANALYSIS
Helpful
to achieving the objective
Harmful
to achieving the objective
Internal origin
(attributes of the organization)
External origin
(attributes of the environment)
S
Strengths
W
Weaknesses
O
Opportunities
T
Threats

Figure 8: *The four components of SWOT in a 2 × 2 matrix.*

15 Static and Dynamic review

15.1 Static Testing

Static Testing is a type of software testing in which software application is tested without code execution. Manual or automated reviews of code, requirement documents and document design are done in order to find the errors. The main objective of static testing is to improve the quality of software applications by finding errors in early stages of software development process.

Reviews, the first step in static testing, can be conducted in numerous ways. Reviews are performed to find and remove errors found in supporting documents. This process can be carried out in four different ways:

- Informal: informal reviews will not follow any specific process to find errors. Coworkers can review documents and provide informal comments.
- Walkthrough: the author of whichever document is being reviewed will explain the document to their team. Participants will ask questions, and any notes are written down.
- Inspection: a designated moderator will conduct a strict review as a process to find defects.
- Technical/peer reviews: technical specifications are reviewed by peers in order to detect any errors.

15.2 Dynamic Testing

Dynamic Testing is a term used in software engineering to describe the testing of the dynamic behavior of code.

In dynamic testing the software must actually be compiled and run. It involves working with the software, giving input values and checking if the output is as expected by executing specific test cases which can be done manually or with the use of an automated process.

Dynamic testing is performed at different levels. Let's look at the basic definition of what each testing means.

- Unit testing: the testing done by the developers right after they build the code to ensure their code is working as they expected and according to the requirements.
- System testing: type of testing the QA department does after the code has been built and this covers the entire system/application in terms of testing. The QA team does thorough testing to make sure the functionality of the system is in-line with the requirements.
- Integration testing: is a joint effort (in most companies) by the QA and development teams to make sure each individual module (after being tested) then connected to other modules or systems is still working as expected. This is basically the testing for the entire workflow from start to end.
- User Acceptance Testing: is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

Static Testing	Dynamic Testing
Testing was done without executing the program	Testing is done by executing the program
This testing does the verification process	Dynamic testing does the validation process
Static testing is about prevention of defects	Dynamic testing is about finding and fixing the defects
Static testing gives an assessment of code and documentation	Dynamic testing gives bugs/bottlenecks in the software system
Static testing involves a checklist and process to be followed	Dynamic testing involves test cases for execution

Table 2: *Difference between Static and Dynamic Testing.*

16 Reflection on 1. Assignment (IEEE 830 standard)

In the first assignment of the course, the professor gave us an assignment description about a hypothetical real case in which the company Rema 1000 would like to implement a calculator that would calculate the price of the buyers' basket.

In order to transform the idea into something real and tangible, it is recommended to apply the practice for Software Requirements Specifications IEEE 830 standard.

This is a recommended practice for writing software requirements specifications. It describes the content and qualities of a good software requirements specification (SRS) and presents several sample SRS outlines.

This recommended practice is aimed at specifying requirements of software to be developed but also can be applied to assist in the selection of in-house and commercial software products. However, application to already-developed software could be counterproductive.

When software is embedded in some larger system, such as medical equipment, then issues beyond those identified in this recommended practice may have to be addressed.

This recommended practice describes the process of creating a product and the content of the product. The product is an SRS. This recommended practice can be used to create such an SRS directly or can be used as a model for a more specific standard.

The benefits of applying IEEE 830 standard to projects is enormous, however these are some of the specific benefits:

- Establish the basis for agreement between the customers and the suppliers on what the software product is to do.
- Reduce the development effort.
- Provide a basis for estimating costs and schedules.
- Provide a basis for estimating costs and schedules.
- Facilitate transfer.
- Serve as a basis for enhancement.

Establishing these requirements before starting to develop the project helps in the short and long term to avoid delays and manage the project much better.

17 Reflection on 2. Assignment (IEEE 829 standard)

Assignment two consisted of applying the IEEE 829 standard to the Rema 1000 calculator according to the requirements established in assignment one.

This standard describes a set of basic test documents that are associated with the dynamic aspects of software testing (i.e, the execution of procedures and code). The standard defines the purpose, outline, and content of each basic document. While the documents described in the standard focus on dynamic testing, several of them may be applicable to other testing activities (e.g., the test plan and test incident report may be used for design and code reviews).

The standard addresses the documentation of both initial development testing and the testing of subsequent software releases.

The standard does not call for specific testing methodologies, approaches, techniques, facilities, or tools, and does not specify the documentation of their use. The standard also does not imply or impose specific methodologies for documentation control, configuration management, or quality assurance.

However, the teacher provided us with a small template to fill in and together with the IEEE 829 standard manual we could apply these tests to the assignment.

18 Conclusion

In conclusion, after having put into practice the concepts acquired during the course and the manuals read, it has been possible to put the theory into practice and see very good results.

Starting and managing a project from scratch to seeing it real is a long road full of ups and downs, problems and a lot of confusion. Having a great knowledge of the standards used in the software industry gives a great advantage to those who know how to apply them.

Therefore, software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function, saving time and problems to all the people involved in the project and provides a basis for product enhancement.

Applying these standards together with software tests allow to deliver high-quality products with the fewest possible errors.

In addition, applying both IEEE 829 and IEEE 830 standards help the basis for agreement between the acquirers or suppliers on what the product is to do and serves as a basis for software improvement making projects that do not apply these standards have a great disadvantage against projects that are using them.

19 References

Tutorials Point. *CMMI Maturity Levels.*
https://www.tutorialspoint.com/cmmi/cmmi-maturity-levels.htm

Microsoft. *Background to Capability Maturity Model Integration (CMMI).*
https://docs.microsoft.com/en-us/azure/devops/boards/work-items/guidance/cmmi/guidance-background-to-cmmi?view=azure-devops

Sync-resource. *A Breakdown of CMMI.*
https://sync-resource.com/basics-of-cmmi/

Plays-in-Business. *SCAMPI: Standard CMMI Appraisal Method for Process Improvement.*
https://www.plays-in-business.com/scampi-standard-cmmi-appraisal-method-for-process-improvement/

Wikipedia. *Standard CMMI Appraisal Method for Process Improvement.*
https://en.wikipedia.org/wiki/Standard_CMMI_Appraisal_Method_for_Process_Improve

ment

Mindtools. *Lewin's Change Management Model.*
https://www.mindtools.com/pages/article/newPPM_94.htm

Systematic. *Systematic is one of the best companies worldwide at delivering software on time.*
https://systematic.com/en-gb/our-company/how-we-work/cmmi5-documented-high-level-of-maturity/

TMMI. *TMMI Model.*
https://www.tmmi.org/tmmi-model/

Wikipedia. *Project management.*
https://en.wikipedia.org/wiki/Project_management

BBC. *The SMART principle of goal setting.*
https://www.bbc.co.uk/bitesize/guides/zxd9j6f/revision/2

GeeksforGeeks. *Software Engineering - SDLC V-Model.*
https://www.geeksforgeeks.org/software-engineering-sdlc-v-model/

Wikipedia. *V-Model (software development).*
https://en.wikipedia.org/wiki/V-Model_(software_development)

Scrum. *WHAT IS SCRUM?.*
https://www.scrum.org/resources/what-is-scrum

Scrum guide. *The 2020 Scrum Guide.*
https://scrumguides.org/scrum-guide.html

Kanbanize. *What Is Kanban? Explained for Beginners.*
https://kanbanize.com/kanban-resources/getting-started/what-is-kanban

Atlassian. *What is a kanban board?.*
https://www.atlassian.com/agile/kanban/boards

ScienceDirect. *Unified Process.*
https://www.sciencedirect.com/topics/computer-science/unified-process

Wikipedia. *SWOT analysis.*
https://en.wikipedia.org/wiki/SWOT_analysis

Investopedia. *Strength, Weakness, Opportunity, and Threat (SWOT) Analysis.*
https://www.investopedia.com/terms/s/swot.asp

Guru99. *Static Testing vs Dynamic Testing: What's the Difference?.*
https://www.guru99.com/static-dynamic-testing.html