

CSSWENG T2 AY22-23

Introduction to Software Engineering

GOAL

Transform you from being a
novice to becoming a
professional

Software Project Outcomes

Successful

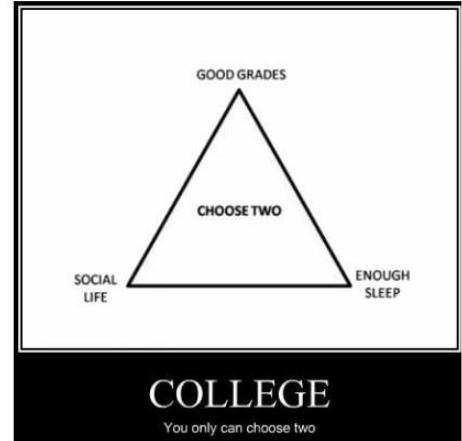
on budget, on time, high customer satisfaction and value

Challenged

over budget, late, mediocre customer satisfaction and value

Failed

cancelled prior to completion, delivered but never used



Different members have a different understanding of the project's goals and requirements.



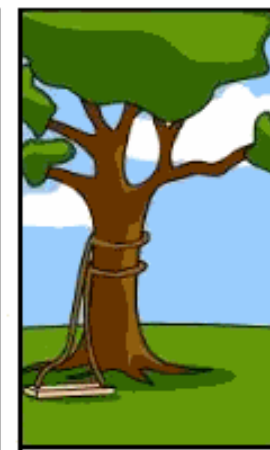
How the customer explained it



How the Project Leader understood it



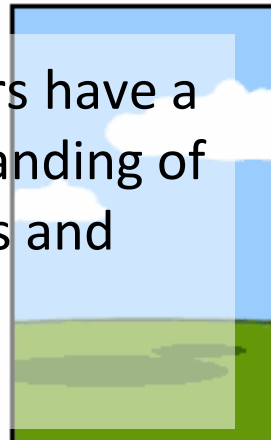
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



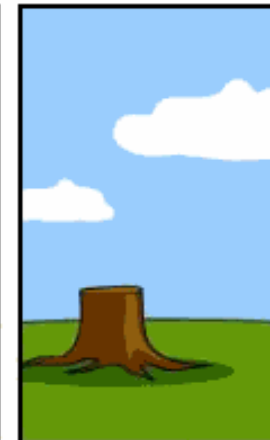
How the project was documented



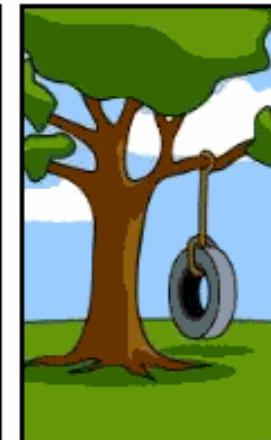
What operations installed



How the customer was billed



How it was supported



What the customer really needed

The practice of

Software Engineering

Software

A set of **computer programs** and **associated documentation** (requirements, design models, test cases, user guides)

[Sommerville, 2016]

Engineering

the **design** and **building** of complex products

[Merriam-Webster]

Software Engineering is

The **process** of creating high
quality software solutions in the
least amount of **time** and **effort**
possible.



Stages of Software Development

Remember engineering is a design!

Stages of Software Development

Requirements Engineering



Software Design



Software Construction



Software Verification & Validation



Software Maintenance

Software Process

- a **systematic** approach in SE
- Sequence of **activities** that leads to the production of a software product



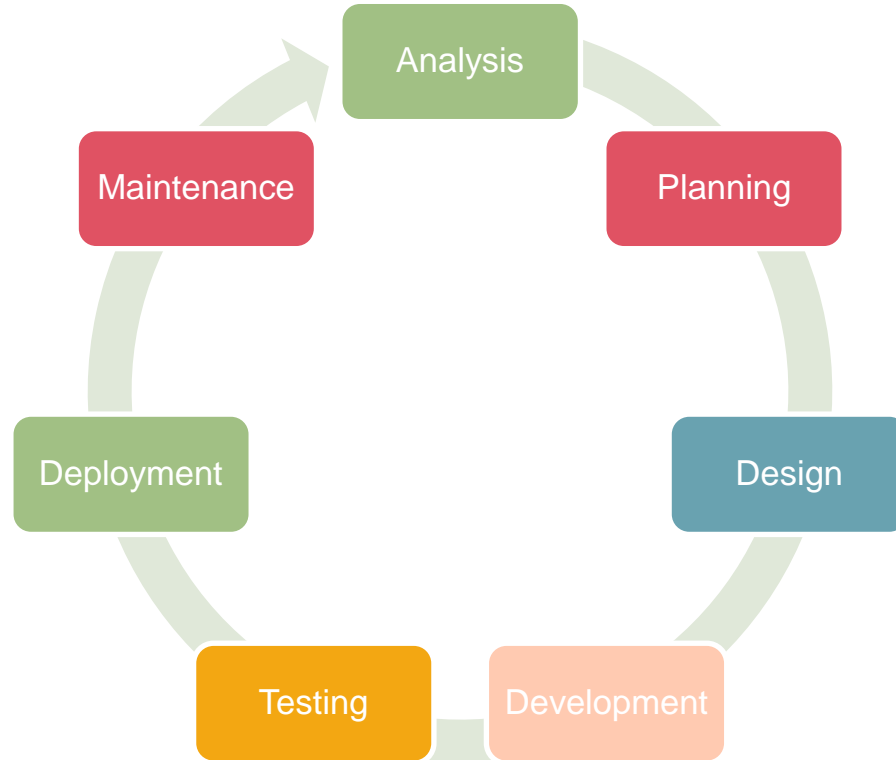
Software Development Lifecycle (SDLC)

A description of the steps that should be performed
when building software products

WHAT IS A SDLC?

- ❖ Software Development Life Cycle is a **framework** for software development process.
- ❖ **Structured processes** for creating high-quality and low-cost software in the shortest possible time.
- ❖ The goal is to **meet** and **exceed** customer expectation and demands.

PHASES



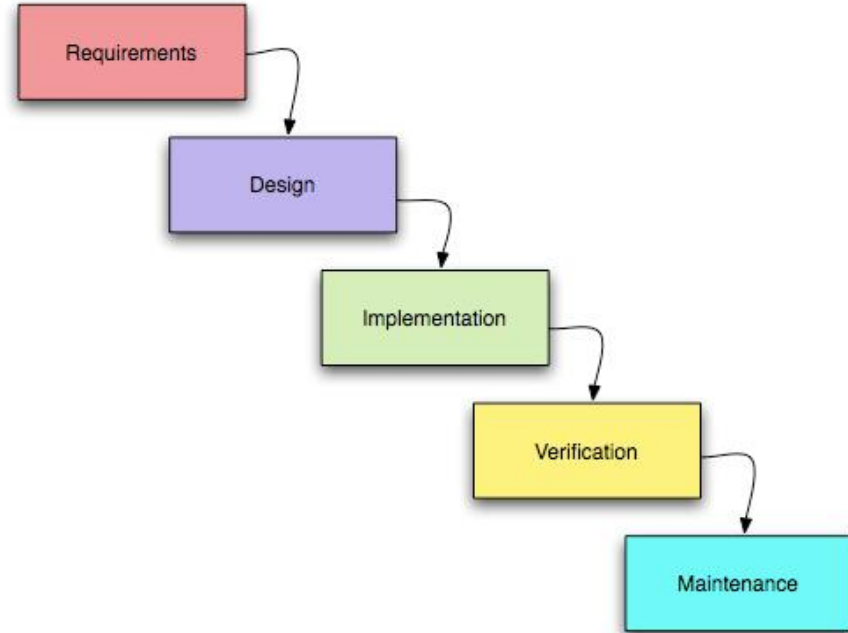
SDLC Models

Who can do it faster?

https://jamboard.google.com/d/182zjaISxihXn2fjRO4wi9j5c_iqOGMrPXECLUcUcpvk/edit?usp=sharing

Waterfall

- Design before coding, analysis before design
- Documentation is important in every step
- Disciplined
- Document and test in every phase
- Client only sees the product at the end

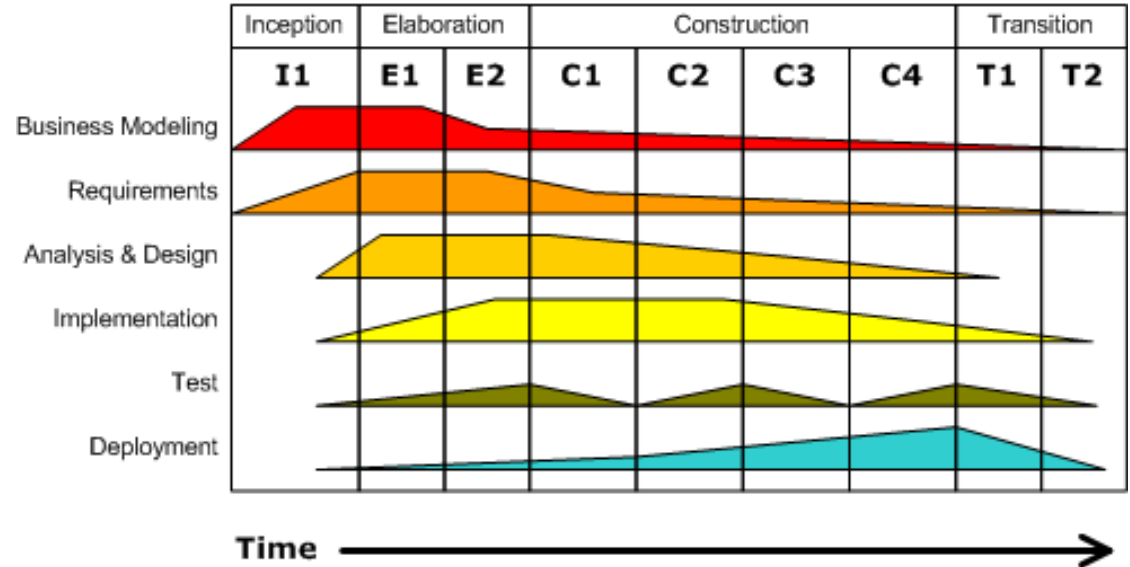


Iterative

- We have a moving target
- Feature creep
- Sometimes synonymous to **Spiral Model**

Iterative Development

Business value is delivered incrementally in time-boxed cross-discipline iterations.



Model

Incremental

- Stepwise refinement
- Incremental builds
- Evolutionary software

HOW **NOT TO BUILD** A MINIMUM VIABLE PRODUCT



1



2



3



4

ALSO HOW **NOT TO BUILD** A MINIMUM VIABLE PRODUCT



1



2



3



4

HOW **TO BUILD** A MINIMUM VIABLE PRODUCT



1



2



3



4

Model

Agile

- agility
- usability
- reliability
- reusability

AGILE MANIFESTO

- INDIVIDUALS & INTERACTIONS
OVER PROCESSES & TOOLS
- WORKING SOFTWARE OVER
COMPREHENSIVE DOCUMENTATION
- CUSTOMER COLLABORATION OVER
CONTRACT NEGOTIATION
- RESPONDING TO CHANGE OVER
FOLLOWING A PLAN

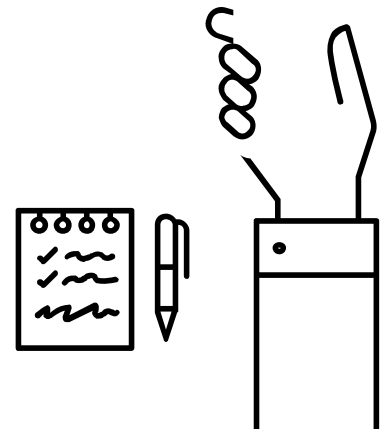
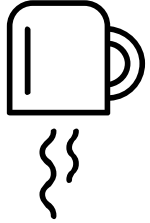
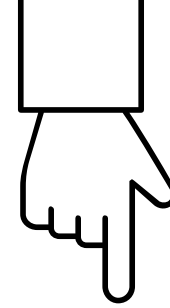
THE

processing

What do you think are the project characteristics that we need to consider in choosing a SDLC for a project?

How to choose?

- ▶ Cost and budget
- ▶ Team size
- ▶ Ability to take risks
- ▶ Flexibility
- ▶ Timeline
- ▶ Client and stakeholder collaboration



Waterfall vs Agile:

**When do you choose
one over the other?**

Waterfall vs Agile

Advantages

Waterfall Model

- Software Planning and Design is **straightforward**.
- Progress is easily **measured**.
You know exactly where you are in the development process.

Agile Model

- Clients are **heavily involved** (this can also be a con).
- A working (albeit incomplete) version of the software can be **more quickly** delivered.

Waterfall vs Agile

Disadvantages

Waterfall Model

- Requirements are **not easy** to obtain completely at the start of a software project.
- Clients only see the software **close to the end** of the process, which could lead to **lower customer satisfaction**.

Agile Model

- Requires **discipline**. Because of its inherent flexibility, members of the team must be committed to the project and must enforce standards.



Questions?

Kindly ask them in the chat.

Agile Software Development

Agile Manifesto

We value...

1. Individuals and interactions over process and tools
2. Customer collaboration over contract negotiation
3. Working software over comprehensive documentation
4. Responding to change over following a plan

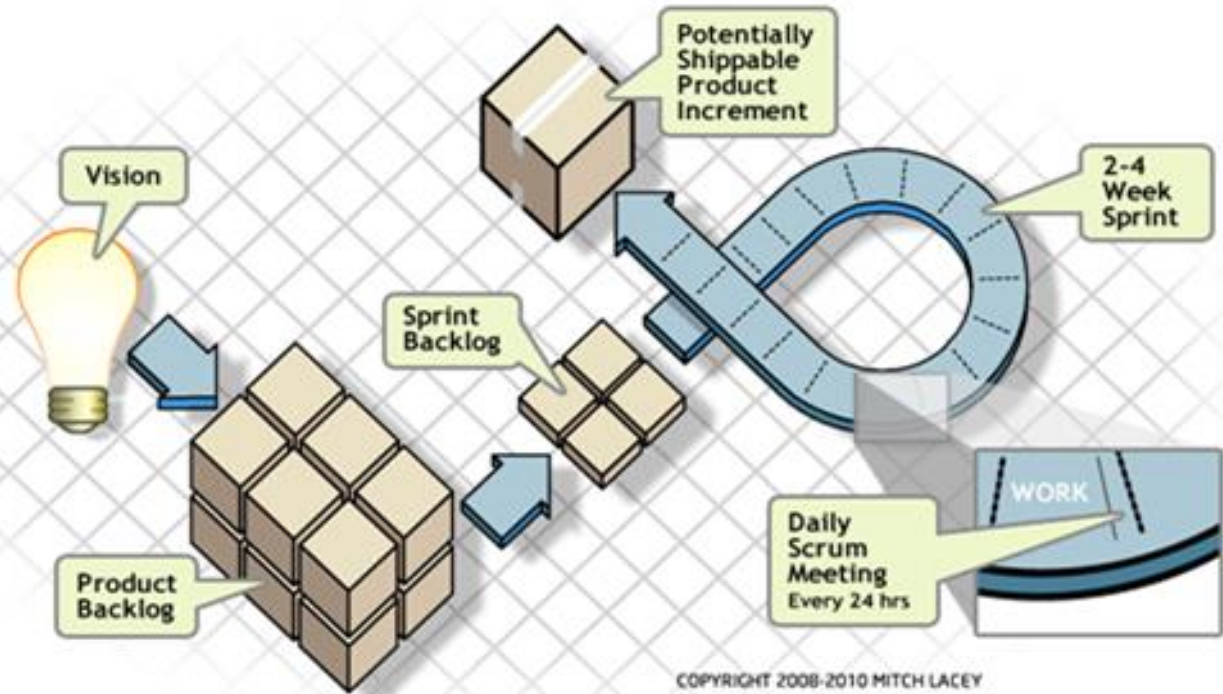
A high-angle, top-down photograph of a scrum in progress on a green rugby field. The scrum is composed of players from two teams: Argentina, wearing light blue and white horizontally striped jerseys, and South Africa, wearing solid black jerseys. The players are tightly packed together, pushing against each other. Some players' jersey numbers are visible, such as 6, 9, 8, 7, and 2. The word "SCRUM" is overlaid in large, white, sans-serif capital letters in the center of the image. The background shows the green grass of the field and a white line.

SCRUM

Scrum Agile Method

A framework for
organizing agile projects

Provides **external
visibility** of what is going
on



Scrum Team

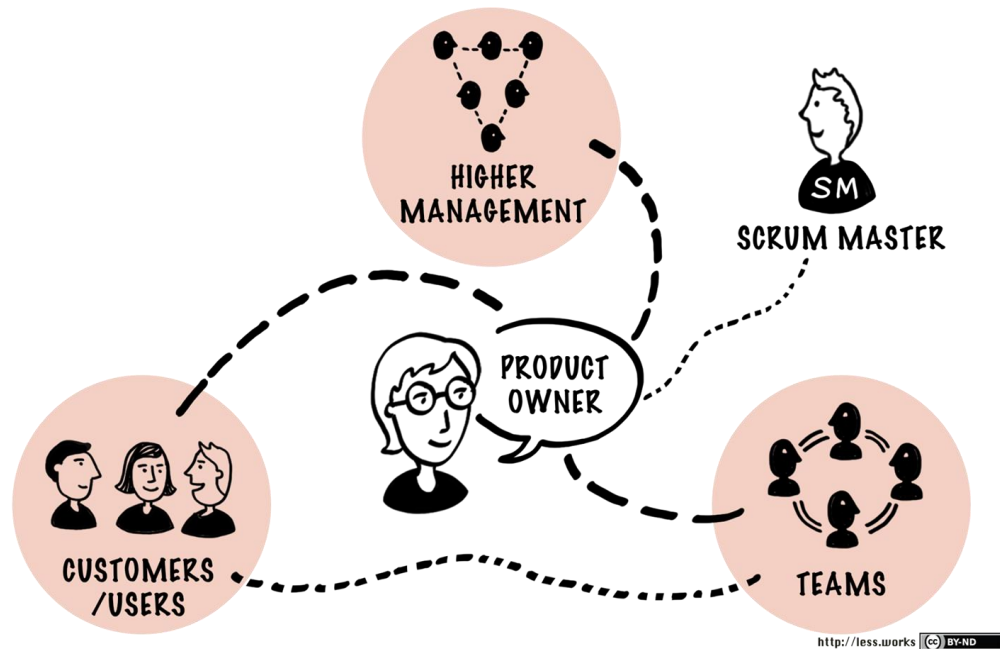
A Scrum Team is composed of the following roles, but the people assigned to these roles are not fixed.

- Product Owner
- Scrum Master
- Developers
- Designer
- Quality Assurance/Tester



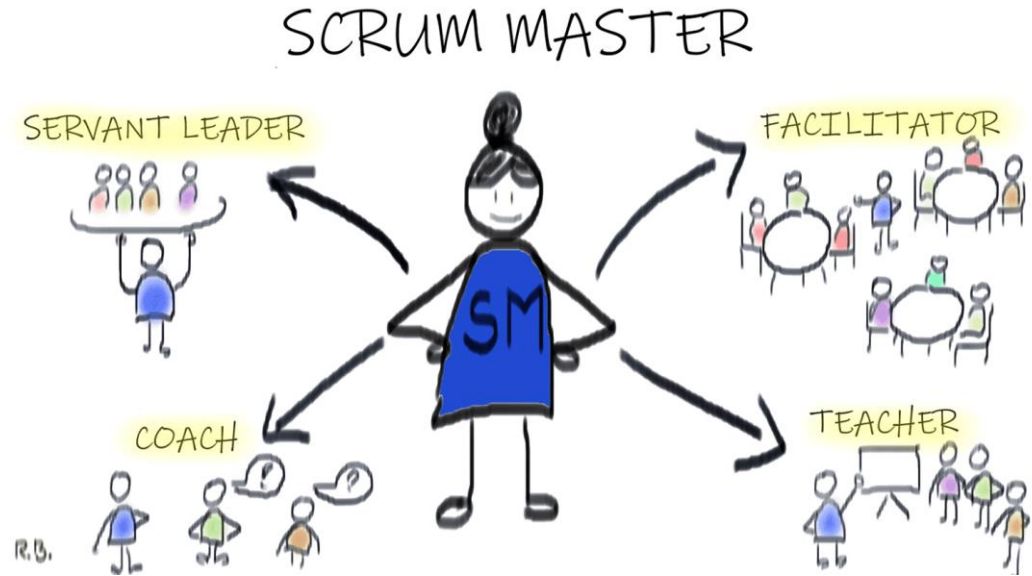
Scrum Team - Product Owner

- The product owner is one person, not a committee
- Voice of the customer
- Manages the product backlog
- Maximizes the value (business value) of the product that the team delivers



Scrum Team - Scrum Master

- Similar to a **project manager**
- Accountable for removing impediments that hinder the team in delivering the product goals and deliverables
- Ensures the team adheres to the scrum theory



Scrum Team - Developers

- Carry out all the work required to build increments of value every sprint
- Create a releasable increment of the “Done” product, having a size that should be small enough to remain nimble but large enough to complete work
- The developers are cross-functional. As a team they have the skills necessary to create a product increment.
- Accountability belongs to the team as a whole (even when each developer has a specific area of focus)

Scrum Team - Designer & Quality Assurance

- The designer and quality assurance (QA) **roles are not distinguished in Scrum.**
- Designers work on the designing and even developing the user interface and overall user experience.
- *The product owner takes up most of the designer role.*
- QAs develop test plans and automate tests to efficiently assess the correctness and validity of the software solution.
- The QA team is part of the development team.

A high-angle photograph of a scrum in progress during a rugby match. Players from Argentina, wearing light blue and white striped jerseys, are on the left, and players from South Africa, wearing black jerseys, are on the right. They are all huddled together, pushing against each other. The text "SCRUM Terms" is overlaid in the center in a large, white, sans-serif font. The background is a dark green grass field with white yard lines.

SCRUM Terms

SPRINT

The heart of Scrum

It is a time-box of one month or less during which a “Done”, usable, and potentially shippable product increment is created.

MINIMUM VIABLE PRODUCT

Goal of each Sprint

“Done”

Scrum Events

Sprint Planning

Setting of Sprint Goals at the start of a Sprint

Daily Scrum

A 15-minute time-boxed meeting at the start of every day. Activities to be done within the day are planned in the daily scrum.

Sprint Review

Performed at the end of every Sprint to discuss what was done during the Sprint and to inspect the Product Backlog

Sprint Retrospective

Opportunity for the team to reflect on improvements at the end of every Sprint

Artifacts

Product Backlog

An ordered list of everything that is needed for the project, and is a single source of project requirements. The Product Owner is responsible for maintaining this.

Sprint Backlog

A set of Product Backlog items selected for the Sprint, plus a plan for realizing the Sprint Goal

Increment

The Increment is the sum of all Product Backlog items completed during a sprint and the value of the increments of all previous Sprints.

Definition of Done

- Scrum relies heavily on artifact transparency
- Everyone must have shared understanding of what “Done” is
- Defined by the team
- This will guide the Scrum Team in selecting Product Backlog items during Sprint Planning
- Need for Stories, Sprints, Integration, and Releases



Questions?

Kindly ask them in the chat.

References

- http://upload.wikimedia.org/wikipedia/commons/thumb/a/a2/Avianca_767-200_at_El_Dorado.JPG/800px-Avianca_767-200_at_El_Dorado.JPG
- <http://www.nosoloviajeros.com/imagenes/colombia/transmilenio.jpg>
- <http://files.nireblog.com/blogs1/keniecita/files/celular-2.jpg>
- <http://www.fayerwayer.com/up/2008/06/iphone3g.jpg>
- <http://www.slideshare.net/soreygarcia/ingenieria-de-software-para-dummies>
- <http://www.projectsmart.co.uk/docs/chaos-report.pdf>
- http://www.evilliens.com/images/software_engineering_explained.gif
- <http://sunnyday.mit.edu/accidents/Ariane5accidentreport.html>
- <http://stackoverflow.com/questions/84556/whats-your-favorite-programmer-cartoon>
- <http://stuffthathappens.com/blog/2008/03/05/simplicity/>
- <http://www.troyangrignon.com/dilbertsoftwarerequirements.jpg>
- <http://www.phdcomics.com/comics/archive/phd120804s.gif>
- <http://www.phdcomics.com/comics/archive/phd011406s.gif>
- <http://search.dilbert.com/comic/Software%20Bugs>
- The 2020 Scrum Guide — <https://scrumguides.org/scrum-guide.html>