Software Development Process

Dr. Rodrigo Spínola

TDresearchteam
Technical Debt Research Team

VCU Computer Science
College of Engineering

Let's chat about cooking?

What factors contribute to chaos in a restaurant kitchen?

Consequences:

- Unexpected food quality.
- Inconsistent food quality.
- Increased risk of food contamination.
- Slower service and longer wait times.
- Higher food waste and costs.
- Increased stress among kitchen staff.
- Miscommunication leading to incorrect orders.
- Difficulty in inventory management.
- Potential safety hazards and accidents.
- Difficulty passing health inspections.
- Lower overall customer satisfaction.

Looks better? Why?

Consequences:

- Each person has a specific role.
- Consistent food quality.
- Efficient workflow and faster service.
- Clear communication among staff.
- Reduced food waste and costs.
- Improved safety and fewer accidents.
- Easier inventory management.
- Higher staff morale and teamwork.
- Enhanced cleanliness and hygiene.
- Better adaptability to high-demand periods.
- Increased customer satisfaction.

Agenda

- Software development process
- Why do we need processes?
- Waterfall
- Spiral
- Factors for choosing a specific process model
- Process modeling

Stages in Engineering Software

- Traditional stages in software engineering:
  - Requirements
  - Design
  - Implementation
  - Verification and Validation
  - Maintenance
- In practice, some of these can be further split (e.g., architectural design vs implementation design), and more can be added (e.g., deployment)

What is a Software Development Process?

- Let's revisiting this question in the context of stages
- A software **process model** determines the order of the stages involved in software development and evolution
- It provides answer to the following items:

Do we need a software process?

- Designing simple software (such as homework assignments) has two steps:
  1. Step 1: Think!
  2. Step 2: Code!
- Both steps are creative
  - Programmers are happy doing them

Programmers while reviewing the codes

11

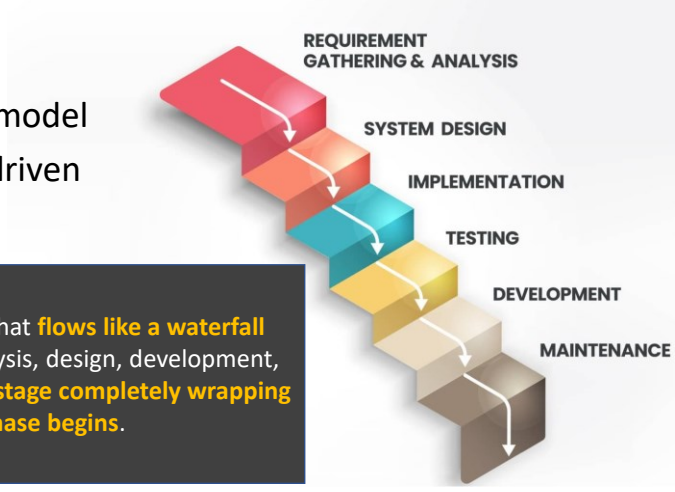
Do we need a software process?

- The two-step process **doesn't scale up with complexity**, for example:
  - How do we split the work among a team of people?
  - How do we ensure we know what the customer meant when they ask for feature X?
  - How do we ensure we give them only what they pay for?
  - What about cross-cutting concerns, such as security, accountability, performance, scalability etc.?

12

Waterfall

- 1st development process model
- Plan and documentation driven



Sequential development process that **flows like a waterfall** through all stages of a project (analysis, design, development, and testing, for example), with **each stage completely wrapping up before the next phase begins**.

Good points about waterfall?

Bad points about waterfall?

15

Waterfall

The Good

- Simple!
- Plenty of documentation, which is good (allows for management of project)
- Still in use since 70s

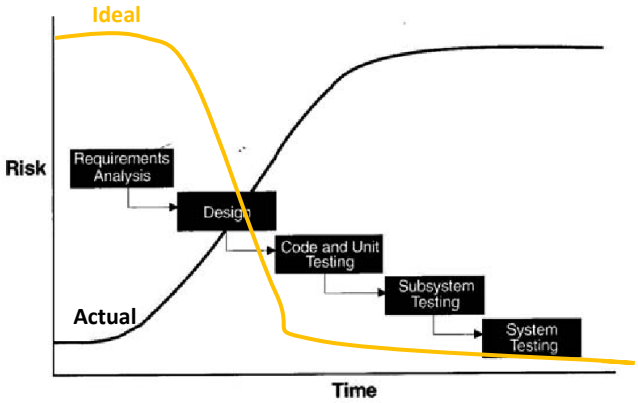
The Bad

- Testing is towards the end of the model, and it may expose fundamental problems, requiring rework
- Low flexibility level makes it difficult to make changes while developing, or even makes it completely impossible
- Customer is not involved

16

Waterfall – The Bad

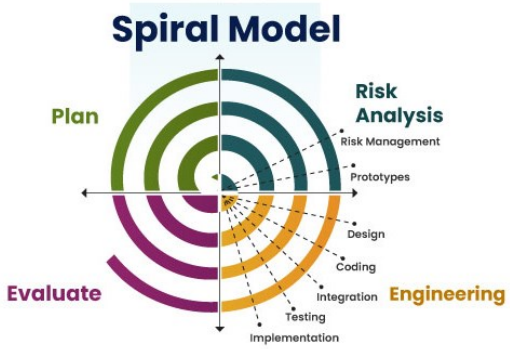
- The customer does not have the opportunity to get acquainted with the system in advance, so he does not see the product until the moment of its completion
- In case it becomes clear in the process of development that the product does not meet market requirements, there will be no room for changes.



17

Spiral Model

- Combine plan-and-document with prototypes
- Rather than plan and document all requirements 1st, develop plan and requirement documents across each iteration of prototype as needed and evolve with the project



18

Spiral Model

The Good

- Iterations involve the customer before the product is completed
  - Reduces chances of misunderstandings
- Risk management part of the lifecycle
- Project monitoring easy
- Schedule and cost more realistic over time

The Bad

- Iterations 6 to 24 months long
  - Time for customers to change their mind
- Lots of documentation per iteration
- Lots of rules to follow, hard for whole Project
- Cost of process is high
- Hard to meet budget and schedule targets

Which is the most appropriate process model to my context?

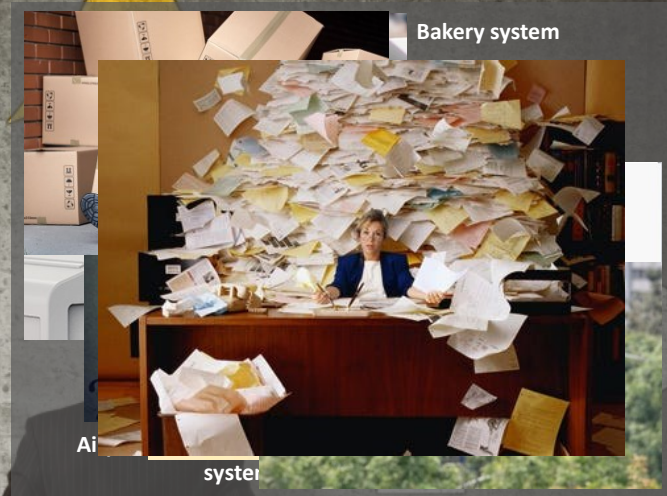
- **There is no silver bullet!**
- So, which factors should I consider when choosing a process model?
  - This is a complex task!
  - There are many of them!

menti.com + 1221 2397

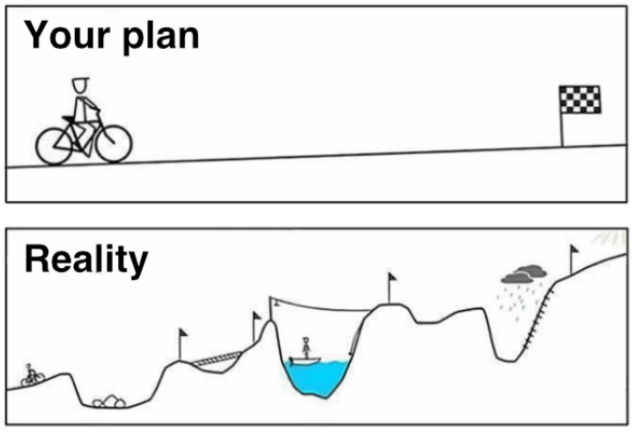
Which factors should I consider when choosing a process model?

Which factors should I consider when choosing a process model?

- Project / application nature
- Team experience and knowledge
- Available tools
- Deliverables of the project
- Culture



Your plan vs. reality



Is there a perfect way to organize software processes?

- **Peres' Law:** "*If a problem has no solution, it may not be a problem, but a fact, not to be solved, but to be coped with over time*."
- *– Simon Peres (winner of 1994 Nobel Peace Prize for Oslo accords)*

Software Development Process

Dr. Rodrigo Spínola

Process Modeling

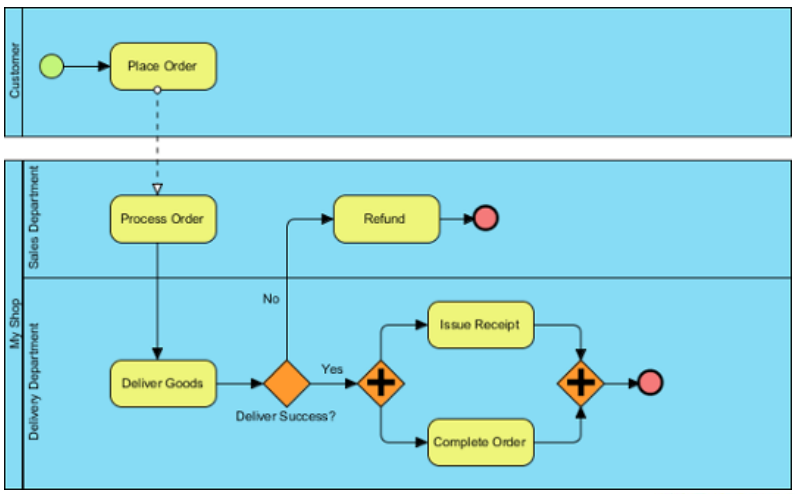
Process Modeling

- Refers to the **graphical representation** of business processes or workflows. Like a flow chart, **individual steps** of the process are drawn out so there is an **end-to-end overview** of the **tasks** in the **process** within the context of the business environment.

Notations

- Activity Diagram from UML
- BPMN (Business Process Model and Notation)
  - is an open standard to diagram a business process
  - it is like a flowchart and uses standardized graphics to represent the participants, choices and flow of the process
  - the diagrams are designed to be detailed, but easy to read without training
    - This allows the same diagram to be used by executives, analysts and technical implementation staff to foster collaboration and understanding between groups

Example - Place Order



Summary

- Software process model determines the order of the stages involved in software development and evolution
- We need processes to scale up with complexity
- There are several types of process models: waterfall, incremental, spiral…
- There is no silver bullet, the context impacts process
- Process modeling helps communicate and share knowledge on how development teams perform their tasks

Software Development Process

Dr. Rodrigo Spínola

TDresearchteam
Technical Debt Research Team

VCU Computer Science
College of Engineering