Software Development Processes

Ruichen cai*
ZJNU
Software Project Management

Abstract

In software engineering, a software development methodology (also known as a system development methodology, software development life cycle, software development process, software process) is a splitting of software development work into distinct phases (or stages) containing activities with the intent of better planning and management. It is often considered a subset of the systems development life cycle. The methodology may include the pre-definition of specific deliverables and artifacts that are created and completed by a project team to develop or maintain an application.[1]

Common methodologies include waterfall, prototyping, iterative and incremental development, spiral development, rapid application development, extreme programming and various types of agile methodology. Some people consider a life-cycle "model" a more general term for a category of methodologies and a software development "process" a more specific term to refer to a specific process chosen by a specific organization. For example, there are many specific software development processes that fit the spiral life-cycle model.

1 Introduction

Why do projects fail?
Review common software project

- Everyone likes to feel that they're making
  progress
- When the team starts to code as soon as the
- project begins, they see immediate gains
- When problems become more complex (as
- they always do!), the work gets bogged down
- In the best case, a team that begins
- programming too soon will end up writing good

Projects can address the wrong needs
Requirements can specify incorrect behavior
Design, architecture and code can be technically flawed
Test plans can miss functionality
The later these problems are found, the more likely they are to cause the project to fail
Managers and teams often want to cut important tasks – especially estimation, reviews, requirements gathering and testing.
If it were faster to build the software without these practices, we would never
use them.
Every one of these practices is about saving time and increasing quality by planning well and finding defects early.

Cutting them out will cost time and reduce quality.Summary
Understand why projects fail and how to improve their chances of success

2 Related Work

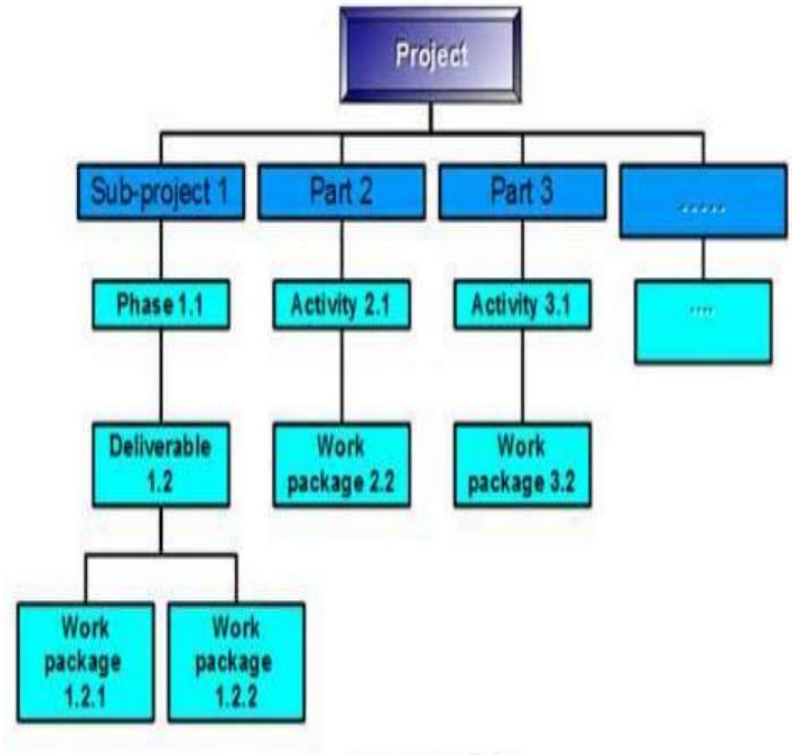
Project Planning Activities
Software Scope and Feasibility
Resources
Software Project Estimation
Decomposition Techniques
Empirical Estimation Models
Who needs software?
Most software is built in organizations for people with specific needs.
A stakeholder is a anyone who has an interest (or stake) in the software being completed
A user is someone who will need to use the software to perform tasks.
Sometimes stakeholders will be users; but often the stakeholder will not use the software.\

3 Overview

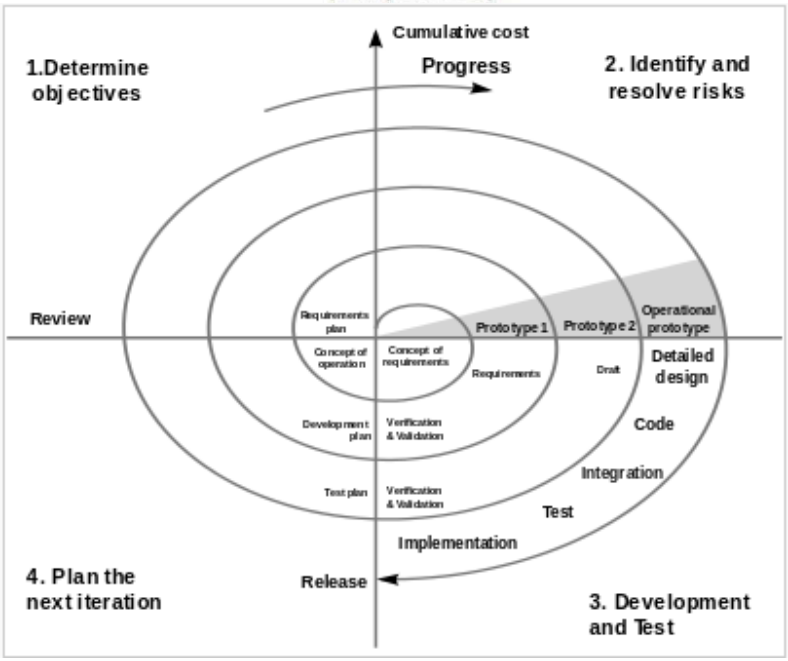
A typical vision and scope document follows an outline like this one:
1. Problem Statement
a) Project background
b) Stakeholders
c) Users
d) Risks
e) Assumptions
2. Vision of the Solution
a) Vision statement
b) List of features
c) Scope of phased release (optional)
d) Features that will not be developedWhat is the Project Plan?
The project plan defines the work that will be done

4 Methods/Techniques

Ready-to-use existing software acquired from third party (COTS) or from (internal) past projects
Full-experience components
• Existing specifications, designs, code, test data from past projects similar to software to be developed (for current project). May require little modifications
Partial experience component
• Existing specifications, designs, code, test data from past projects related to software to be developed (for current project) but will require substantial modifications
New components
• Software components that must be built for current project3. (Development) Environment
resources
These include hardware and software support for a software project
Hardware and software elements availability and time window must be specified



(Example of WBS)



5 Conclusion

All project documents, schedules, estimates, plans and other work products should be shared with the entire team, managers, stakeholders, users and anyone else in the organization who wants them.
Major decisions that are made about the project should be well-supported and explained.Don't second-guess your team

Acknowledgements

I would like to thank my colleagues and reviewers for taking time out of their valuable schedule to help make this report more concise and clearer.