Software Process Models

EECS 3311

Ilir Dema
demailir@yorku.ca

What is a process?

What is a process?

A process is a set of activities that interact to produce a result.

https://en.wikipedia.org/wiki/Process

What is a process?

A process is a set of activities that interact to produce a result.

https://en.wikipedia.org/wiki/Process

design,
build framework,
make essential components,
polish,
paint,
assemble,
road testing,
...

Software development process

What are the activities in software development?

Software development process

Specification

Designing

Implementing

Testing

Maintaining

Software development process

A software process model is a description of the sequence of activities carried out in a SE project, and the relative order of these activities.

Testing

Maintaining

Generic software process models

- **Code-and-fix model**
  - write some code, debug it, repeat until finished
- **Waterfall model**
  - a relatively linear sequential approach
- **Iterative Models**
  - **Prototype model**
    - used to understand/evolve the requirements
  - **Spiral model**
    - a risk-driven model, assesses risks at each step, and does the most critical action immediately
- **Agile model**
  - **Extreme Programming (XP)**
  - **Scrum**

  focus on process adaptability and customer satisfaction, a good fit in fast changing environments

Impact dimensions

By changing the SP model, we can improve and/or tradeoff:

- Development speed (time to market)
- Product quality
- Project visibility
- Administrative overhead
- Risk exposure
- Customer relations
- etc.

Code-and-fix model

- starts with little or no initial planning
- usually no documentation
- anyone who has knowledge of coding can use this

http://dinotimor.blogspot.com/2013/04/the-use-of-software-enterprise.html

Code-and-fix model pros and cons

Low overhead

Low expertise, anyone can use it!

…

Code-and-fix model pros and cons

Low overhead

Low expertise, anyone can use it!

…

No way to assess progress or manage the project

Does not fit a complex project

Unclear what and when will be delivered

Low quality

…

Wh[illegible]kinds of projects can use code-and-fix model?

- Sm[illegible]f of concept”

What kinds of projects can use code-and-fix model?

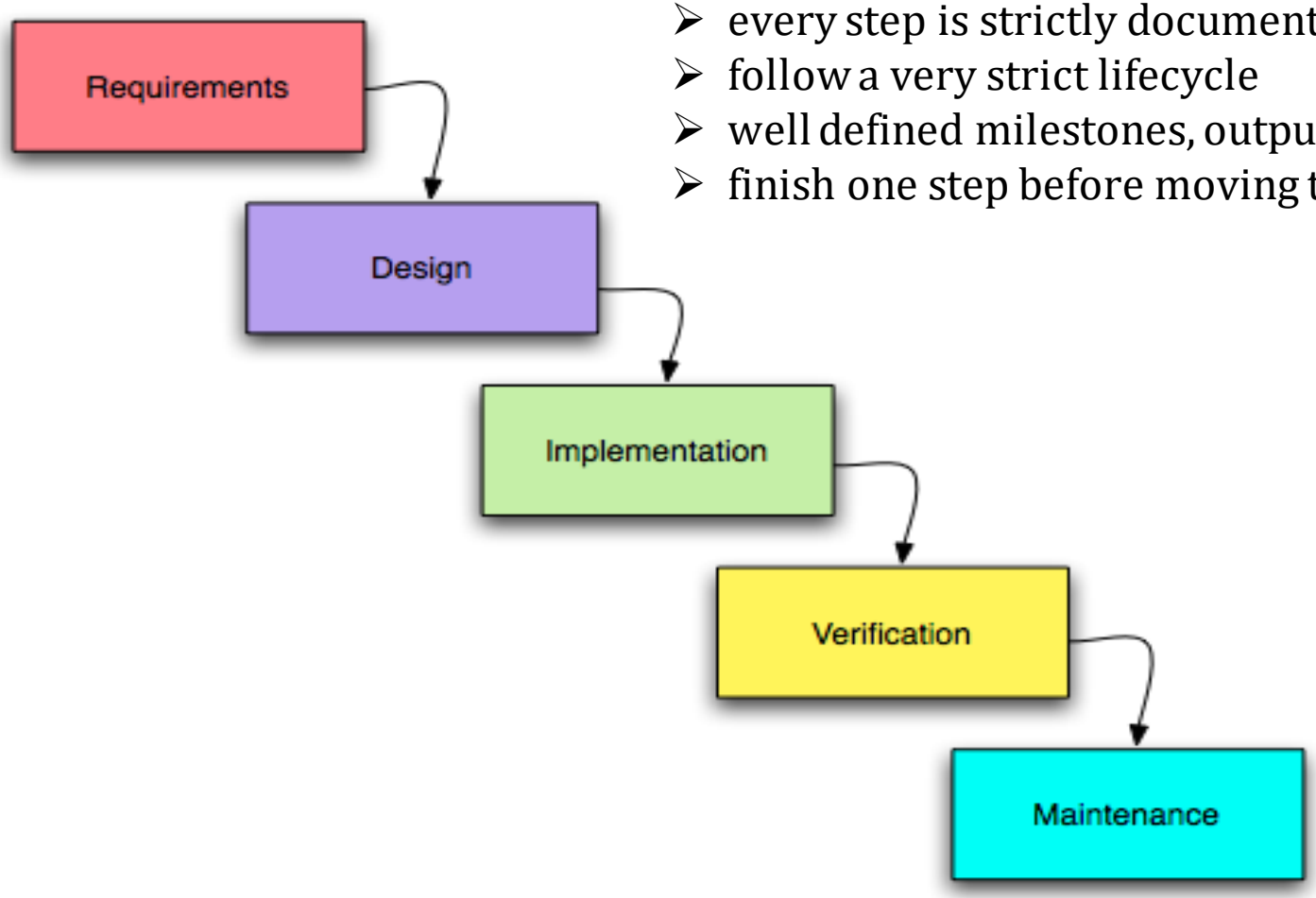
- Sm[illegible]f of concept”
- Your course project ☹

What kinds of projects can use code-and-fix model?

- "Sm[illegible]of of concept"
- Your course project ☹

- **Hacking!** ☺

Waterfall model



- every step is strictly documented and predefined
- follow a very strict lifecycle
- well defined milestones, outputs
- finish one step before moving to the next step

Royce, Winston W. "Managing the development of large software systems", Proc. IEEE Westcon, IEEE Press, 1970. 1970.

Waterfall model pros and cons

Simple to use and understand

Has clear deliverables and milestones

Management is simple

Easy to classify and prioritize tasks

…

Waterfall model pros and cons

Simple to use and understand

Has clear deliverables and milestones

Management is simple

Easy to classify and prioritize tasks

…

Too rigid

Non-adaptive design constraints

Ignores mid-process client feedback

Testing in the very end, which does not give the option of identifying the problem in advance

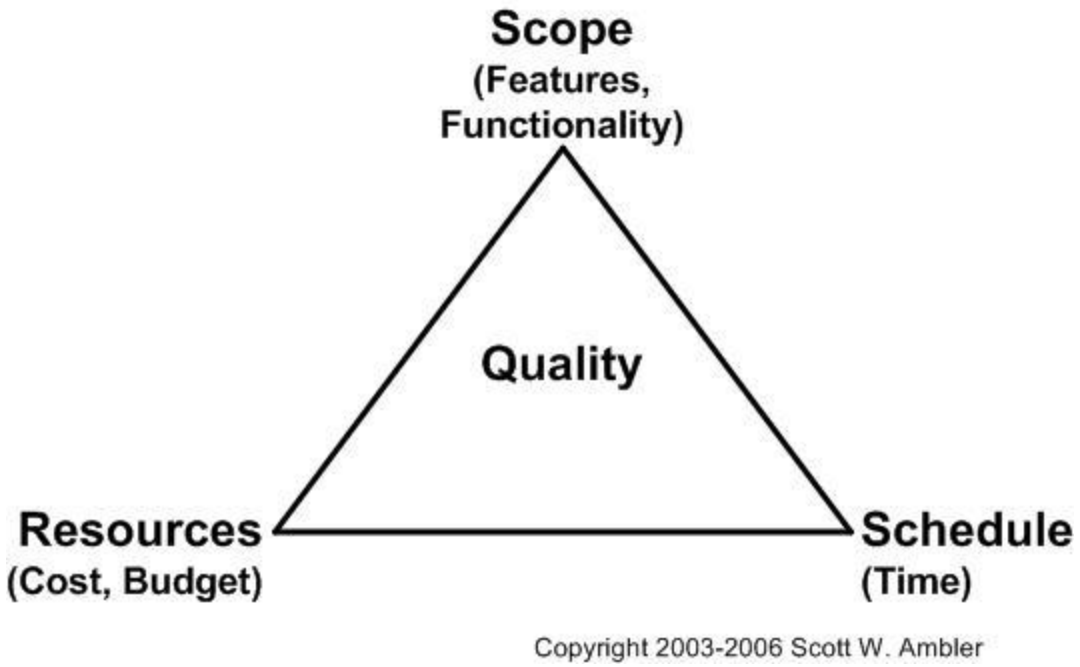
no working software is produced until late during the life cycle

…

Why is this model is not popular now?

- Main assumption: all requirements can be determined at the beginning, no change later on
  - This does not correspond to current software development
  - Data shows change rates from 35%–50% for large projects
  - “change” is the most expensive operation in the waterfall model
  - Feedback (from clients, programmers) comes very/too late
- **Note:** that doesn’t mean you should fall into the opposite extreme and do no planning at all

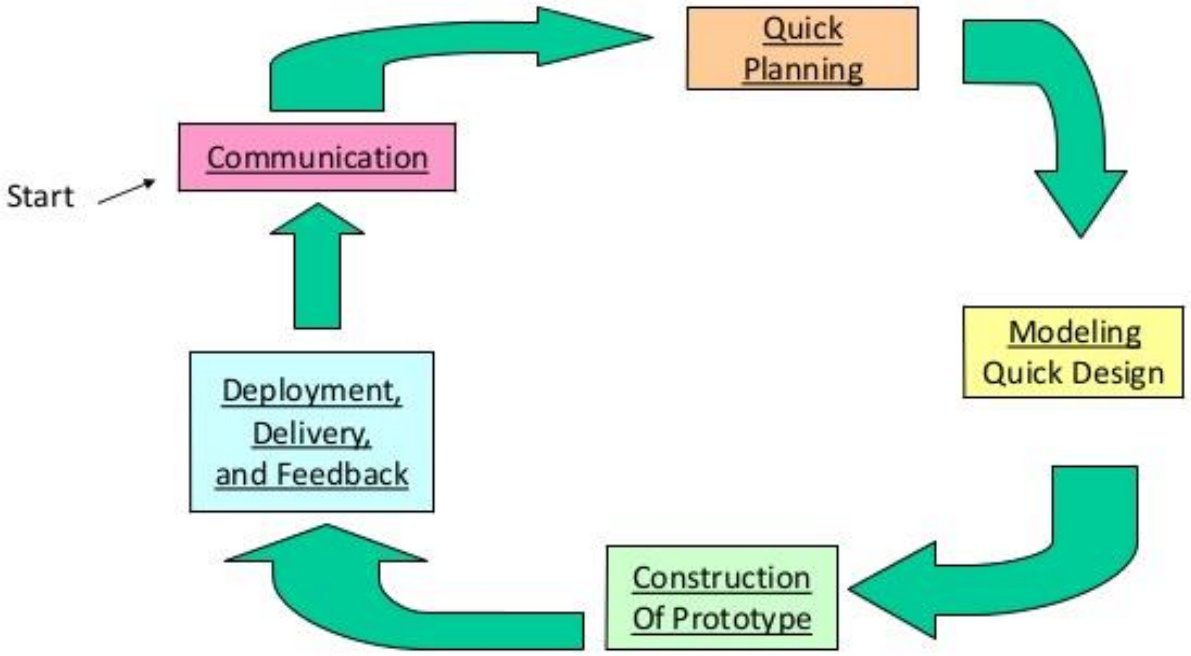
The broken iron triangle



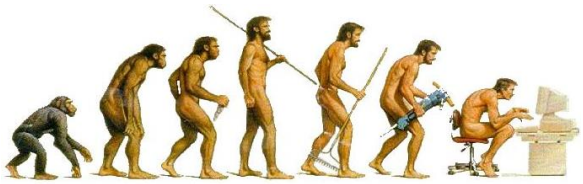
What kinds of projects can use waterfall model?

- Requirements are precisely defined
  - aviation system
  - warehousing systems
- There are no ambiguous requirements
- Product definition is stable, no changes

Prototype model



- help customers and developers understand the system requirements
- build a prototype, adjust the requirements, and revise the prototype until have a clear picture



https://en.wikipedia.org/wiki/Software_prototyping

Prototype model pros and cons

Can help reduce risk of incorrect user requirements

Missing functionality can be identified easily

Projects have a higher visibility

…

Prototype model pros and cons

Can help reduce risk of incorrect user requirements

Missing functionality can be identified easily

Projects have a higher visibility

…

Requires extensive customer collaboration

Difficult to know how long project will last, users may change requirements now and then

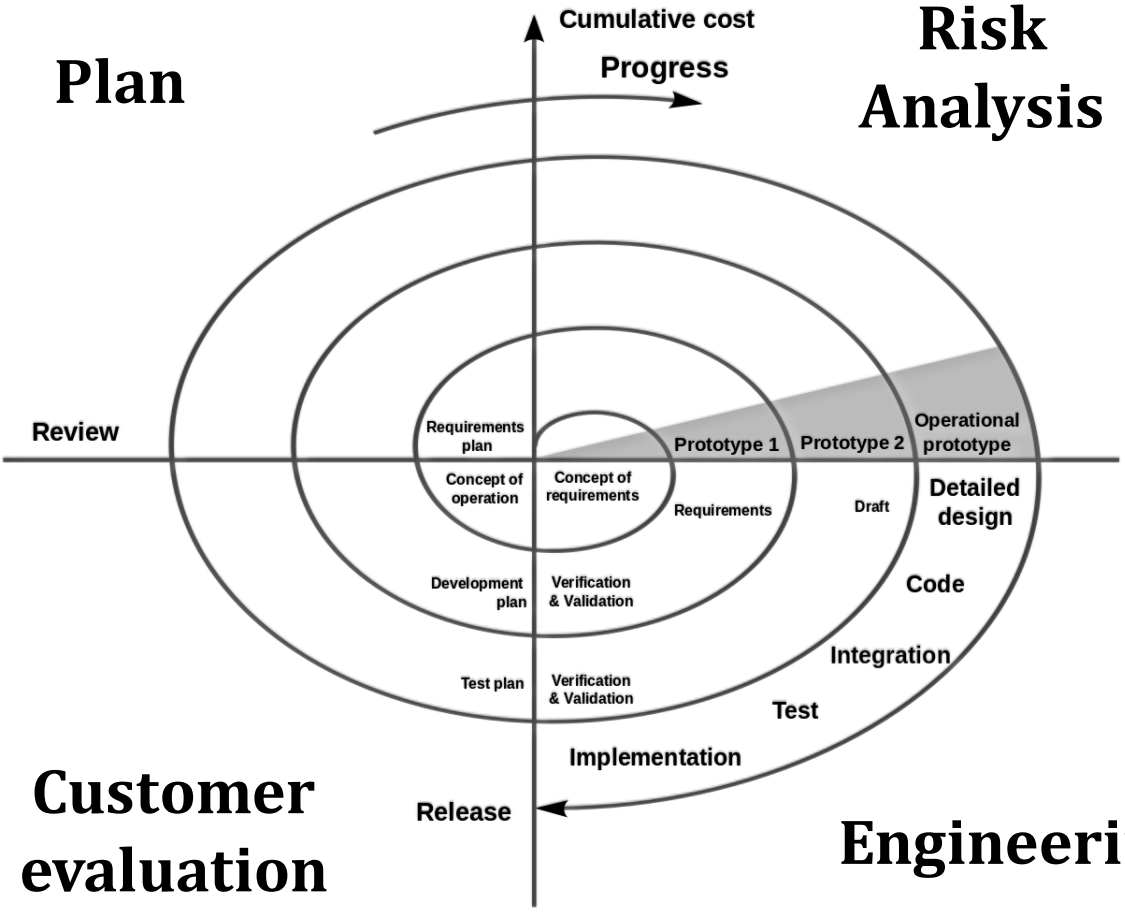
Easy to fall back into code-and-fix without proper design, customer feedback evaluation

…

What kinds of projects can use prototype model?

- When requirements are unclear
- When the users are involved

Spiral model



- a combination of waterfall and prototype models
- includes risk analysis and risk management
- each iteration identifies and solves the sub-problems with the highest risk

Boehm, Barry W. "A spiral model of software development and enhancement." *Computer* 21.5 (1988): 61-72.

Spiral model phases

- **Plan**
  - communicate with the clients to get the necessary requirements
- **Risk Analysis**
  - identify all the possible risks and alternative solutions
- **Engineering**
  - develop/test the software
- **Customer evaluation**
  - customers/clients evaluate the developed system.

Risk analysis

Example Risks
Personnel shortfalls
Unrealistic schedules and budgets
Developing the wrong software functions
Developing the wrong user interface
Continuing stream of requirement changes
Shortfalls in externally furnished components
Shortfalls in externally performed tasks
Real-time performance shortfalls
…

Boehm, Barry W. "A spiral model of software development and enhancement." *Computer* 21.5 (1988): 61-72.

Spiral model pros and cons

Has risk and cost management

More flexible, changing requirements can be handled

Allows for extensive use of prototypes

High visibility, users see the system early

…

Spiral model pros and cons

Has risk and cost management

More flexible, changing requirements can be handled

Allows for extensive use of prototypes

High visibility, users see the system early

…

Management is more complex

Large number of intermediate stages, which requires excessive documentation

Could be expensive for small projects

Relies on developers to have risk-assessment expertise

…

Wh[illegible]inds of projects can use spiral model?

- I[illegible]ed for large, expensive, and c[illegible]ted systems
- For high-risk projects

Agile models

plan quickly, develop quickly, release quickly, and revise quickly

- a user-centered and practice-based model
- a collection of *practices* based on several *values* and proven software engineering *principles*
- development is divided into small iterations, at the end of each iteration, a working product is delivered

The Core of Agile models

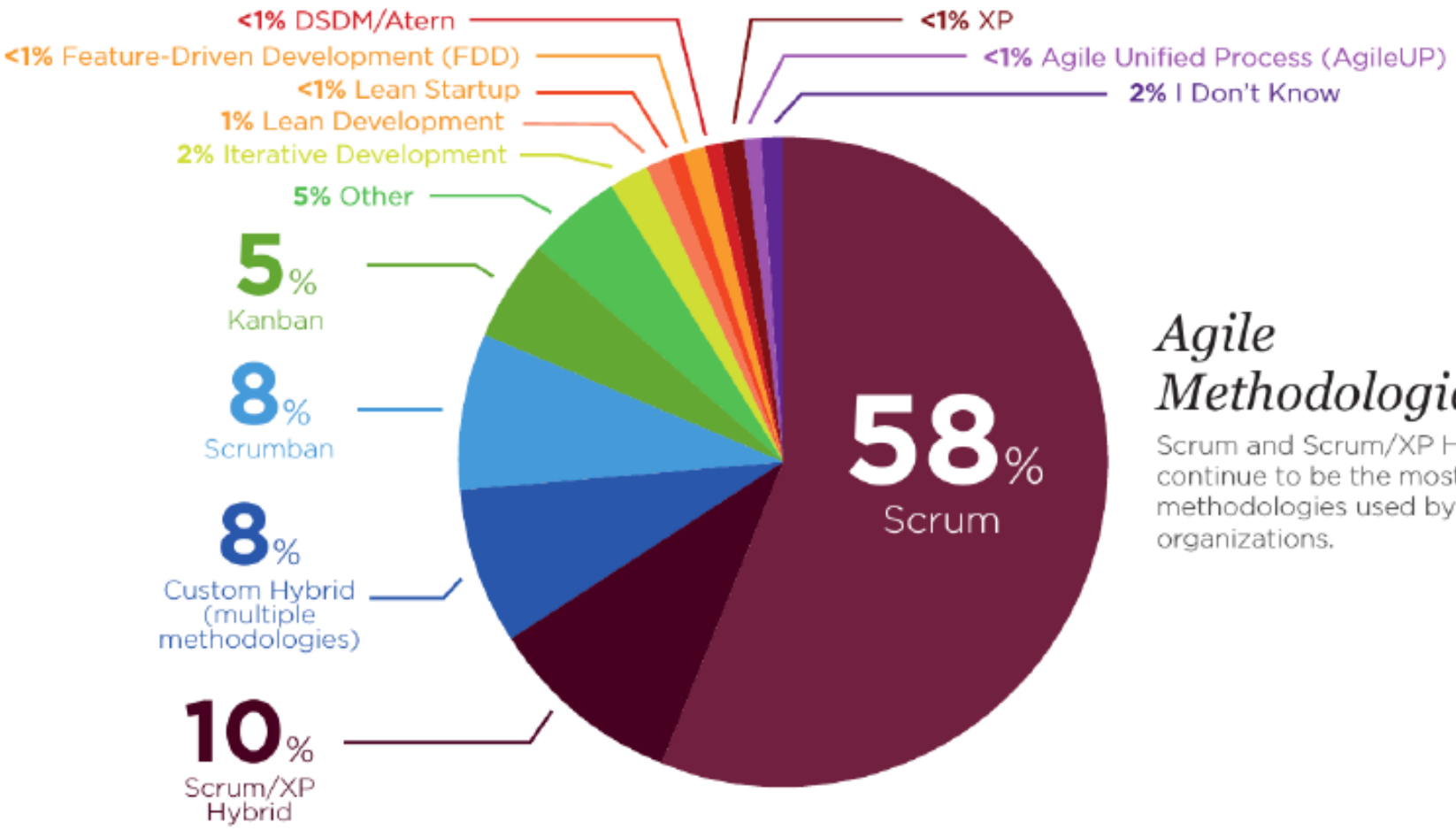
Core Principles:

- Assume Simplicity
- Embrace Change
- Close Communication between company and client Incremental Change
- Model With a Purpose
- Multiple Models
- Maximize Stakeholder Investment
- Quality Work
- Rapid Feedback
- Software Is Your Primary Goal
- Travel Light
- …

Core Practices:

- Test-Driven Development
- Continuous delivery of software
- Continuous collaboration with customer
- Continuous update according to changes
- Continuous integration
- Simple design
- Pair programming
- Sharing the codebase between all or most programmers
- A single coding standard to which all programmers adhere
- …

www.ambysoft.com/scottAmbler.html



Agile Methodologies Used

Scrum and Scrum/XP Hybrid (68%) continue to be the most common agile methodologies used by respondents' organizations.

Extreme Programming (XP)

Emphasises

- Individuals and interactions
  - Over processes and tools
- Working software
  - Over documentation
- Customer collaboration
  - Over contract negotiation
- Responding to change
  - Over following a plan
- software engineering practices > project management

XP Practices

- Programming in pairs
- Test-driven development (TDD)
- Continuous planning, change, delivery
- Shared project metaphors, coding standards and ownership of code

http://www.agile-process.org/

XP (Ad-/Disad-)vantages

Lightweight methods suit small-medium size projects

Produces good team cohesion

Emphasises final product

Iterative Test based approach to requirements and quality assurance

XP (Ad-/Disad-)vantages

- Lightweight methods suit small-medium size projects
- Produces good team cohesion
- Emphasises final product
- Iterative Test based approach to requirements and quality assurance

- More difficult to scale up to large projects where documentation is essential
- Needs experience and skill for not to degenerate into code-and-fix
- Programming in pairs is costly
- Test case construction is a difficult and specialized skill

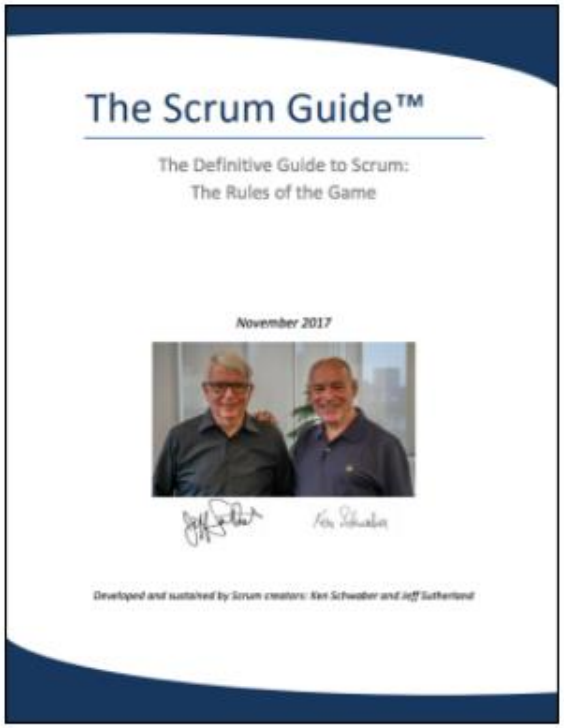
Scrum

A simple process framework for effective team collaboration on complex products.

- Lightweight
- Simple to understand
- Difficult to master

Origins

Takeuchi and Nonaka, 1986: *new approach for commercial product development*



Ken Schwaber and Jeff Sutherland

https://www.scrumguides.org/scrum-guide.html

Scrum has been used by:

- Microsoft
- FB
- Google
- Electronic Arts
- High Moon Studios
- Lockheed Martin
- Philips
- Siemens
- Nokia
- Capital One
- BBC
- Intuit
- Intuit
- Nielsen Media
- First American Real Estate
- BMC Software
- Ipswitch
- John Deere
- Lexis Nexis
- …

Used from small startups to Fortune-100 companies

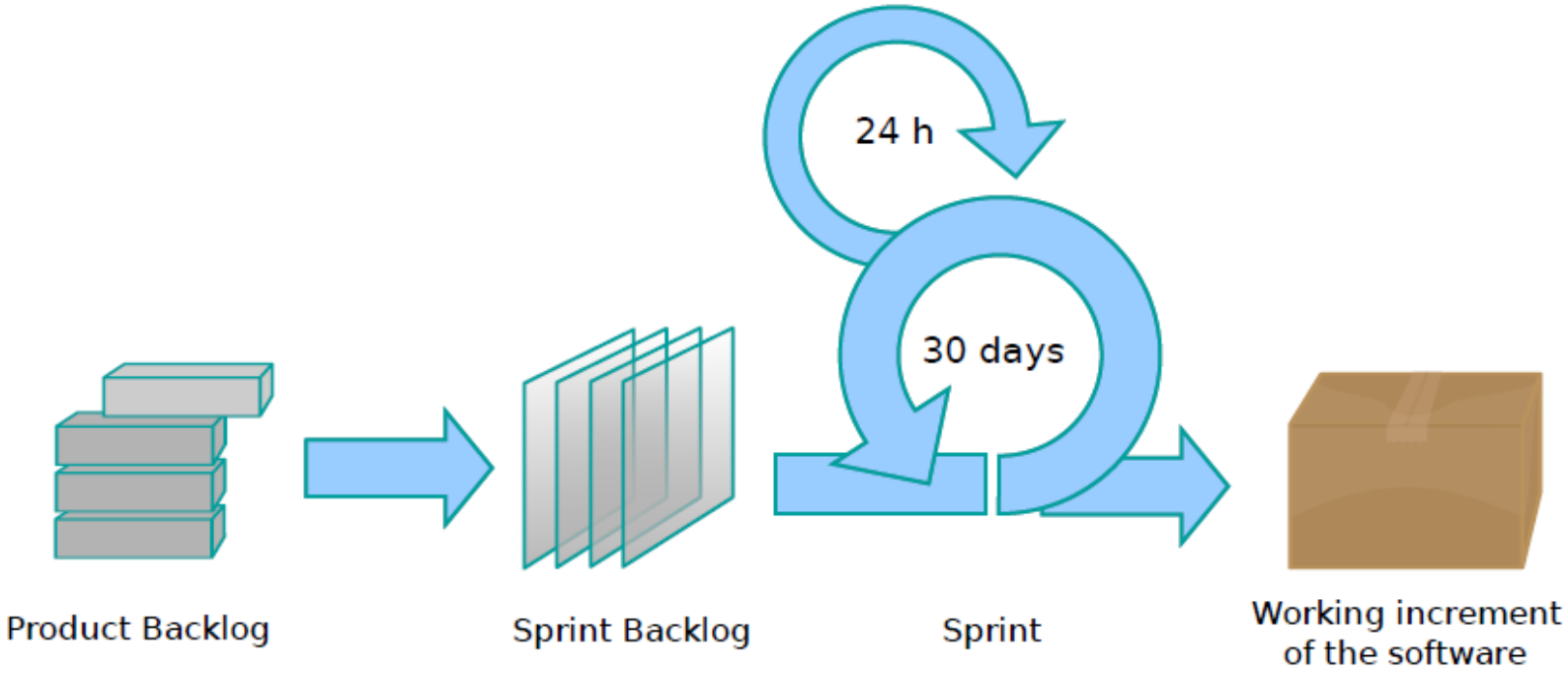
Scrum has been used for:

- Commercial software
- In-house development
- Contract development
- Research
- Financial applications
- ISO 9001-certified applications
- Embedded systems
- 24x7 systems with 99.999% uptime requirements
- the Joint Strike Fighter
- Video game development
- FDA-approved, life-critical systems
- Satellite-control software
- Websites
- Handheld software
- Mobile phones
- Network switching applications
- ISV applications
- Some of the largest applications in use

Characteristics

- Self-organizing teams
- Product progresses in a series of month-long **“sprints”**
- Requirements are captured as items in a list of **“product backlog”**
- No specific engineering practices prescribed
- Uses generative rules to create an agile environment for delivering projects

Scrum overview

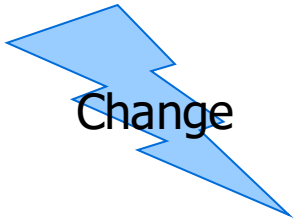


- all changes to be made to the product in future releases
- dynamic & evolves

Sprints

- Scrum projects make progress in a series of "sprints"
  - Analogous to iterations
- Typical duration is 2–4 weeks or a calendar month at most
- A constant duration leads to a better rhythm
- Product is designed, coded, and tested during the sprint

changes during a sprint



- Plan sprint durations around how long you can commit to keeping change out of the sprint

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

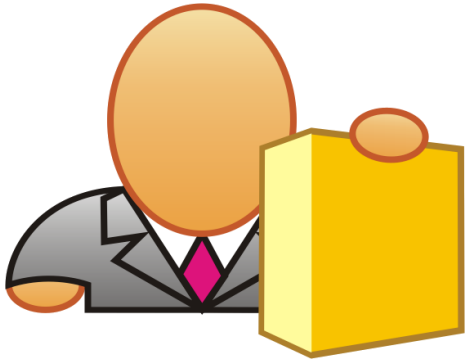
Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Product owner



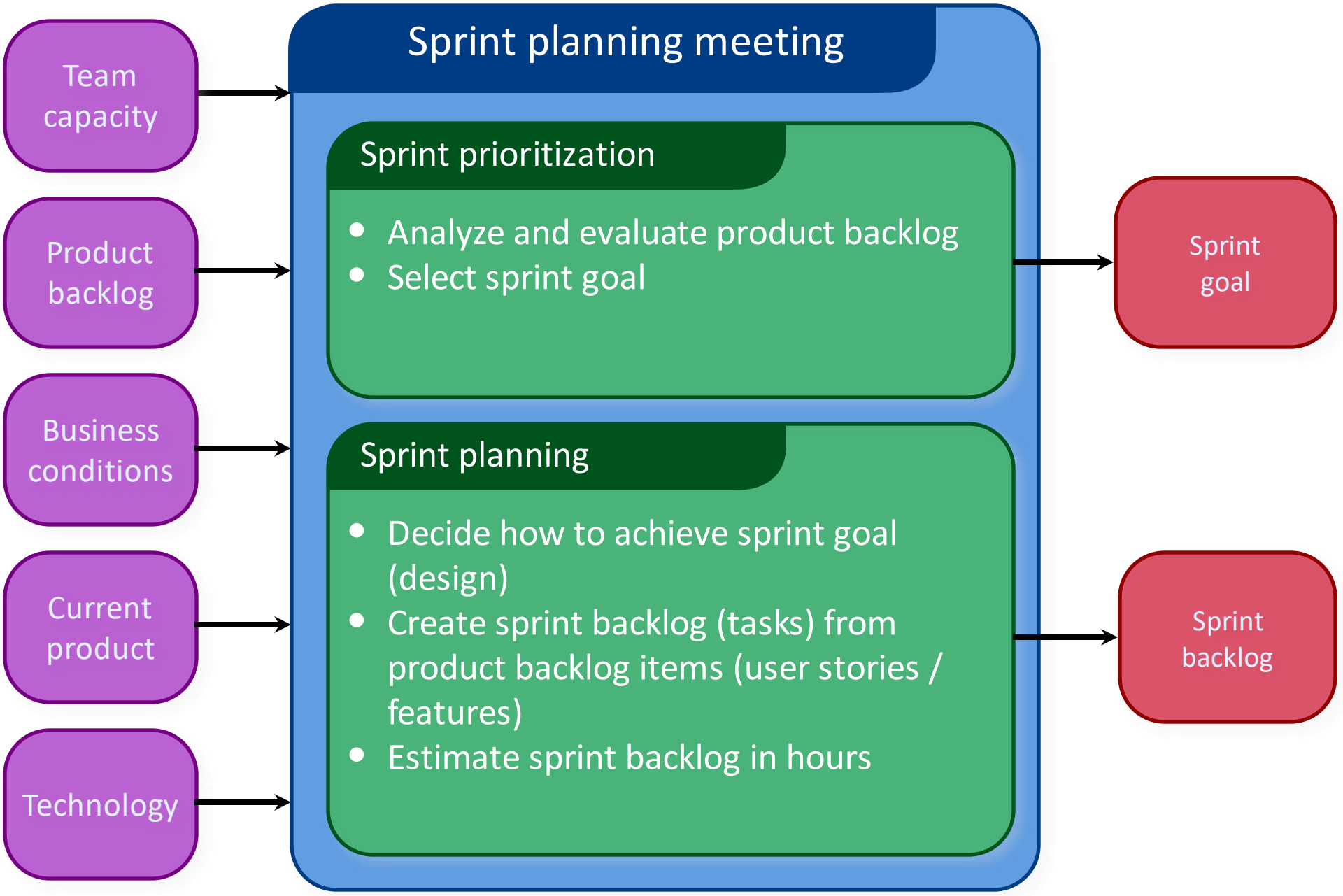
- Define the features of the product
- Decide on release date and content
- Be responsible for the profitability of the product (ROI)
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed
- Accept or reject work results

The ScrumMaster

- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences

The team

- Typically 5-9 people
- Cross-functional:
  - Programmers, testers, user experience designers, etc.


Team capacity
Product backlog
Business conditions
Current product
Technology
Sprint planning meeting
Sprint prioritization
Analyze and evaluate product backlog
Select sprint goal
Sprint planning
Decide how to achieve sprint goal (design)
Create sprint backlog (tasks) from product backlog items (user stories / features)
Estimate sprint backlog in hours
Sprint goal
Sprint backlog

Wh[illegible]kinds of projects can use Agile models?

- P[illegible]hat have dynamic requirements
- When rapid productions are need
  - IT companies

Model Evaluation

- Some criteria for evaluating models
  - Risk Management
  - Quality / cost control
  - Visibility of progress
  - Customer involvement
  - …

https://homes.cs.washington.edu/~stepp/

Model Evaluation

- Some criteria for evaluating models
  - Risk Management
  - Quality / cost control
  - Visibility of progress
  - Customer involvement
  - …
- Overall aim for the one fits your project.

https://homes.cs.washington.edu/~stepp/