Software development process models

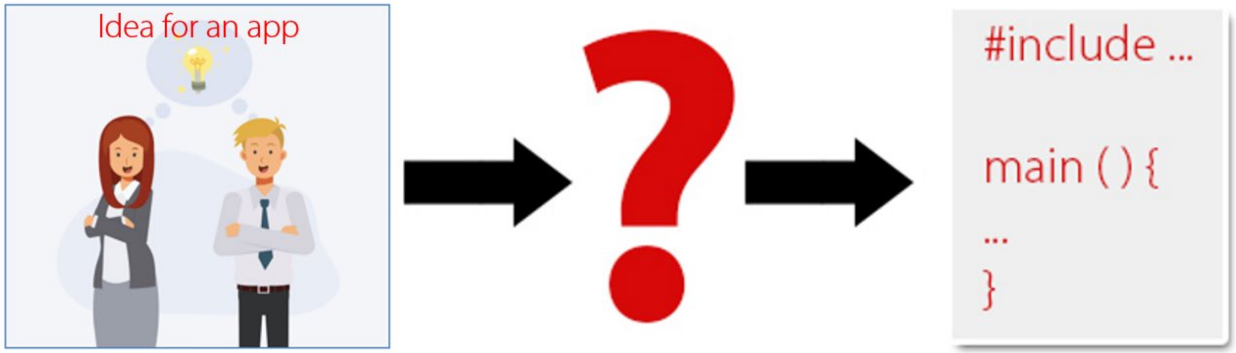
Professor Hossein Saiedian

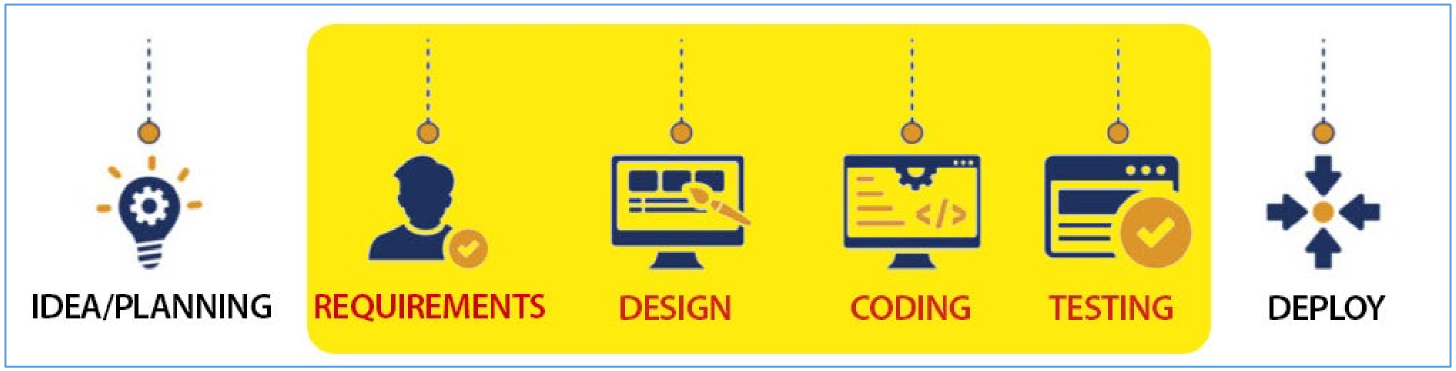
EECS348: Software Engineering

Fall 2024

Software development life cycle

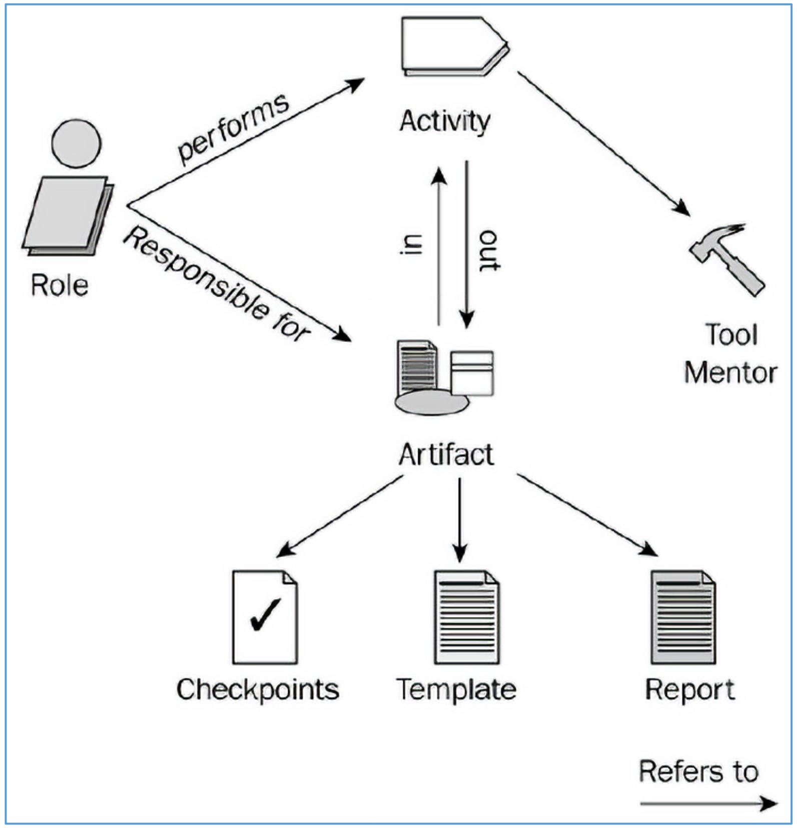
- The software development life cycle
  - A set of processes (a set of related activities)





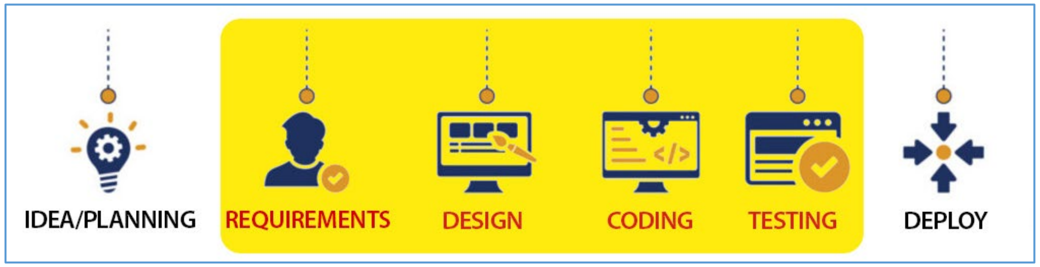
A development process

- A **process** a set of activities that are planned and are performed to achieve a given purpose, and includes
  - Roles and responsibilities
  - Tools
  - Procedures and methods that define how do the tasks and relationship between the tasks



A development process

- Why a process
  - Provides guidelines and a structure
  - Provides for consistency
  - Minimizes redundancies
- **Software development process** a process for building a software
  - Four major activities (very broad)
    - * Requirements engineering
    - * Design
    - * Coding/implementation
    - * Testing



Activities may have different names

- It is possible that an organization uses a different phrase for a set of SE activities
- For example, for requirements engineering, some may say
  - Requirements analysis
  - Requirements definition
  - Requirements gathering
  - System requirements

Activities may be divided

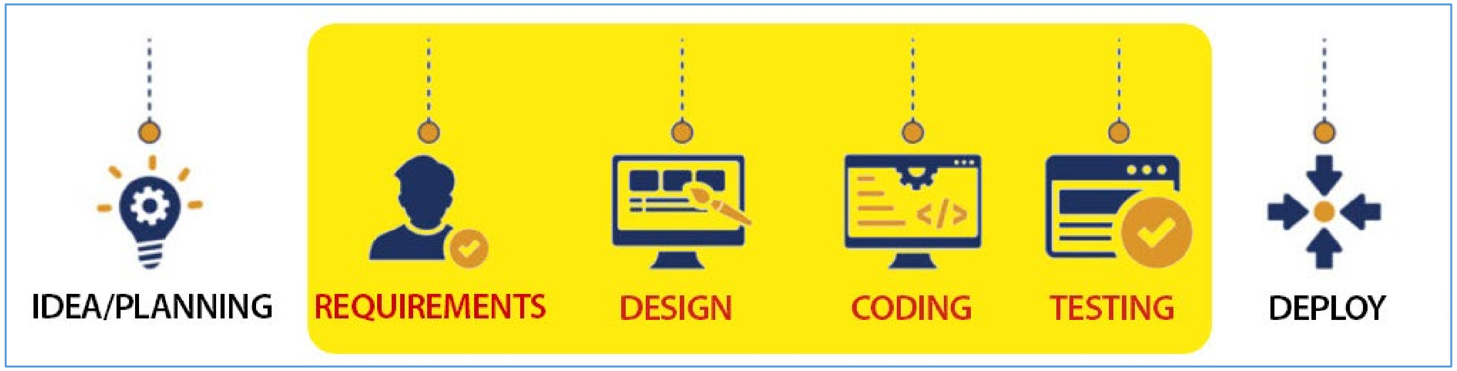
- Software development activities may be presented as a set of distinct activities
- Requirements engineering
  - Requirements elicitation
  - Requirements modeling
  - Requirements specification
  - Requirements validation

Activities may be divided

- Software development activities may be presented as a set of activities
- Design
  - Architectural design
  - Data structure design
  - Component design
  - Interface design

Software development life cycle

- The software development life cycle
- A set of processes (a set of related activities)
- In what order do we do these activities?



Software development models

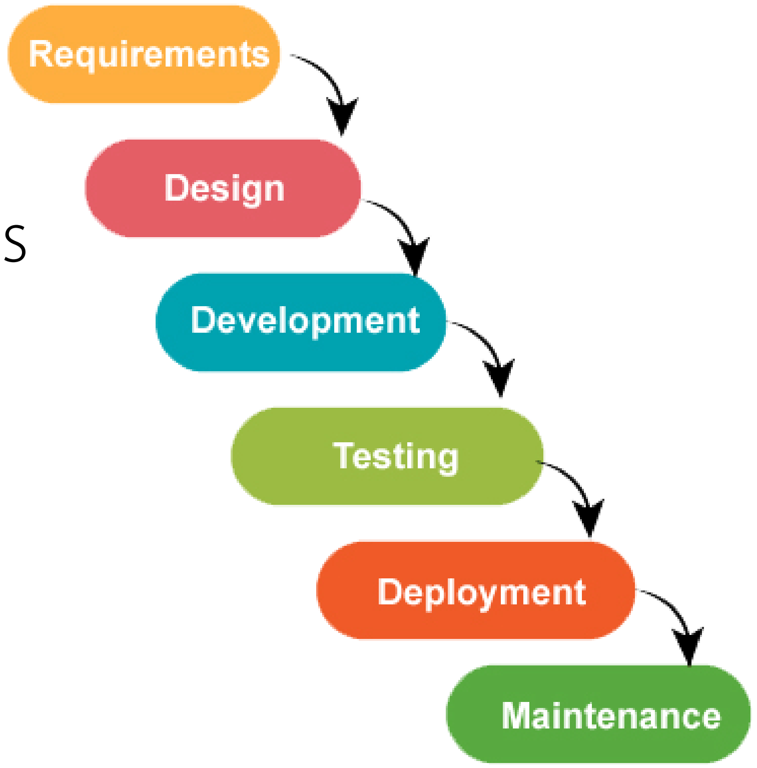
- Describe the ordering of development activities and the expected artifacts (outcomes) from each activity
- Many development models
  - Planned, disciplined, "linear" models
  - Prototyping model
  - Iterative and incremental models
  - Spiral model
  - Agile models
  - …

Software development models

- Describe the ordering of development activities and the expected artifacts (outcomes) from each activity
- Many development models
  - Planned, disciplined, "linear" models
  - Prototyping model
  - Iterative and incremental models
  - Spiral model
  - Agile models
  - …

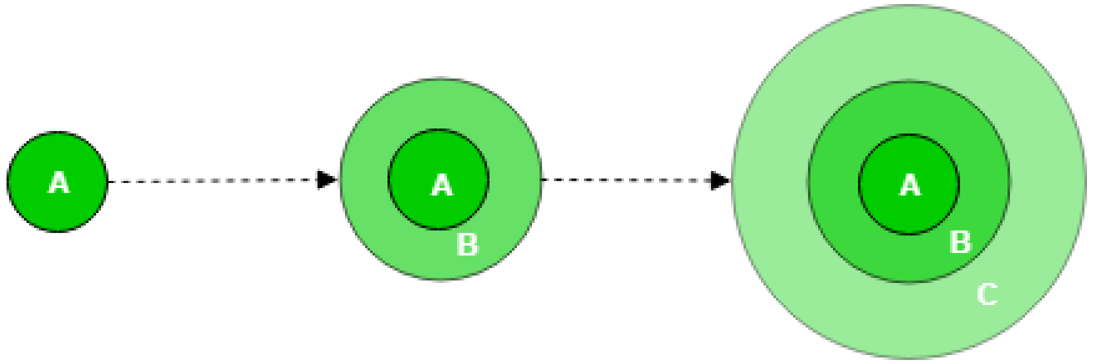
Planned, disciplined models

- AKA the waterfall model
- One of the first process development models proposed
- Works for well understood problems with minimal or no changes in the requirements
- Each major phase is marked by milestones and deliverables (artifacts)
- Long wait before a final product



Incremental and iterative models

- **Incremental** development
  - Starts with a simple working system only with a few basic functionalities
  - Successive versions with additional functionality are implemented and delivered

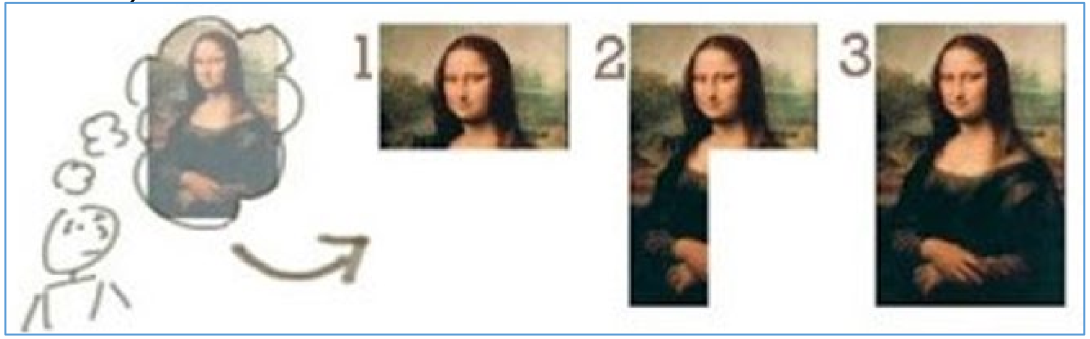


Incremental and iterative models

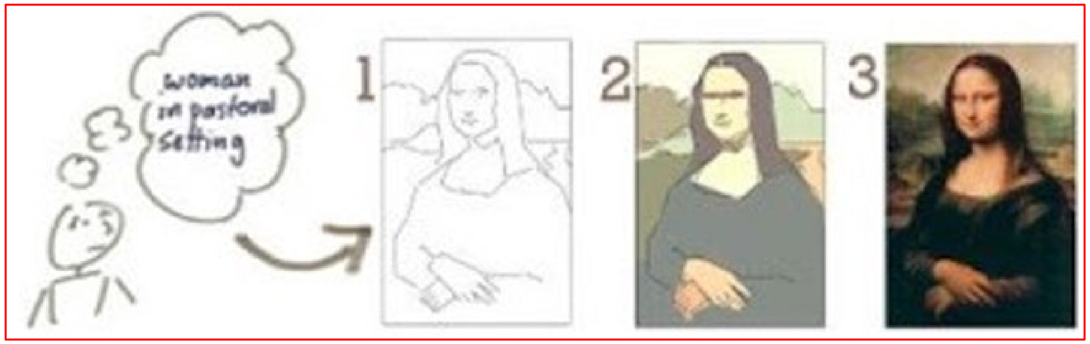
- **Iterative** development
  - The same part of the software is developed repeatedly in cycles, where each cycle is focused on refining and improving the existing functionalities (what already exists not new pieces)
- Example: Consider the development of a search engine
  - First iteration: basic search functionality
  - Second iteration: refine algorithm to improve relevancy
  - Third iteration: optimize code and database accesses
  - *No new features, but the existing features is refined, enhanced*

Incremental and iterative analogy

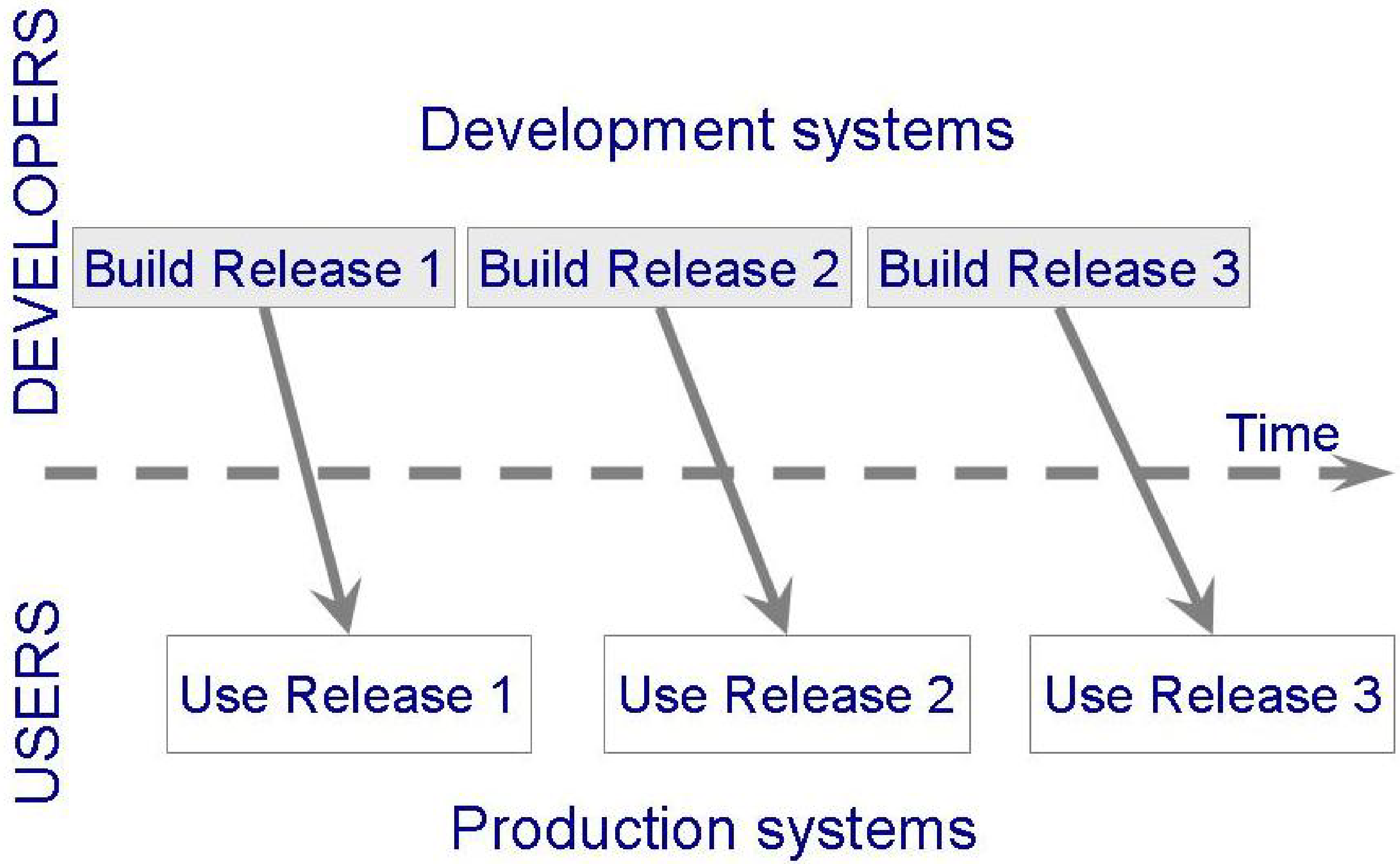
- **Incremental** development: starts with small functionality, and adds new functionality with each new release



- **Iterative** development: starts with full system (but very minimal detail), then enhances the functionality of each software component with each new iteration



Phased development: increments

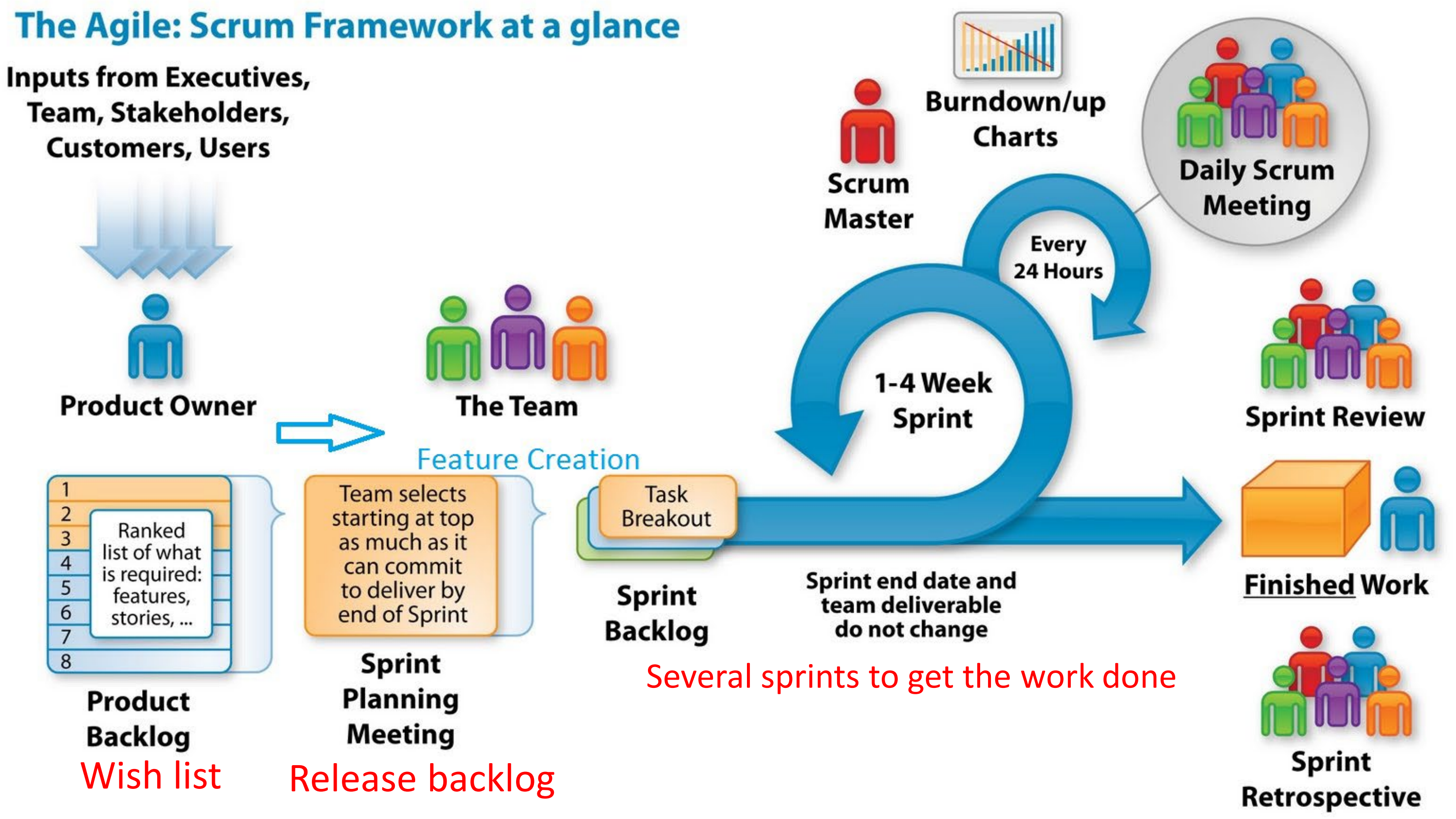


Agile models

Product Backlog: an ordered list of everything that might be needed in the product

Sprint backlog is a subset of the product backlog, consisting of specific items that a team commits to complete in a time-boxed sprint

- Iterative and incremental
  - Several well-known agile models; Scrum is most popular



The Agile: Scrum Framework at a glance
Inputs from Executives, Team, Stakeholders, Customers, Users
Product Owner
The Team
Feature Creation
1
2
3
4
5
6
7
8
Ranked list of what is required: features, stories, ...
Team selects starting at top as much as it can commit to deliver by end of Sprint
Task Breakout
Product Backlog
Sprint Planning Meeting
Sprint Backlog
Scrum Master
Burndown/up Charts
Daily Scrum Meeting
Every 24 Hours
1-4 Week Sprint
Sprint end date and team deliverable do not change
Sprint Review
Finished Work
Sprint Retrospective

Wish list

Release backlog

Several sprints to get the work done

A short intro to Scrum software development

Hossein Saiedian

Based on: Hamid Shojaee's *Intro to Scrum under 10 minutes*

THE UNIVERSITY OF KANSAS

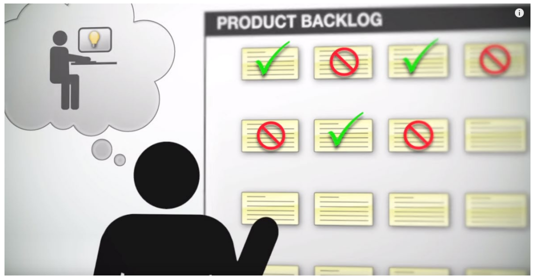
Intro to Scrum

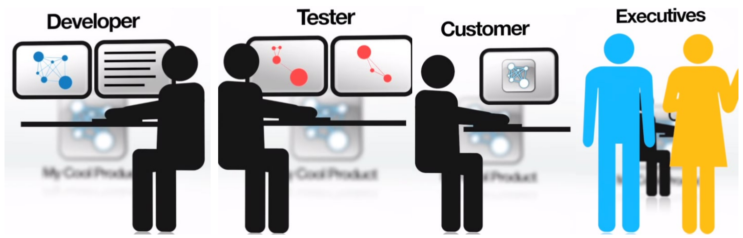
Scrum development methodology

- One of the most popular agile software development methodologies
  - Other ones?
- Important Scrum concepts
  - Team roles
  - Product backlog
  - Sprints
  - Burndown charts
  - Estimation techniques
  - Sprint retrospectives

Scrum roles

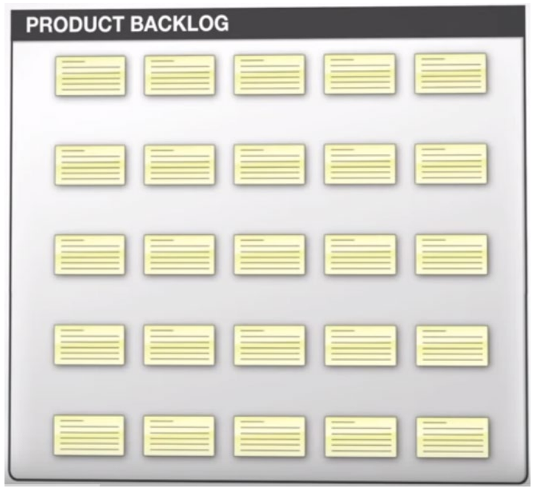
- Product owner
  - Makes sure the right features get into a new release (representing the users, customers)
  - Sets direction
- Scrum master (project manager)
  - Ensures the project progresses smoothly
  - Team members have the tools to get their job done
  - Sets up meetings, monitors, plans releases, …
- Developers, testers, customers, …





Developing your cool new app

- You will get *feature* requests from a variety of stakeholders
  - Features are always from the perspective of the end users
  - Known as user stories: "As a user (role) I want 2F authentication"
  - The role could be a customer, an end users, an executive, …
  - Backlog: A collection of user stories (or wish list)

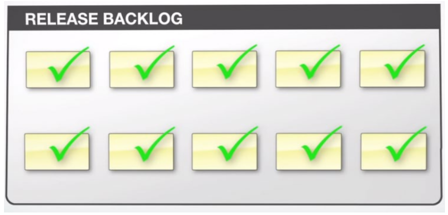


Developing your cool new app

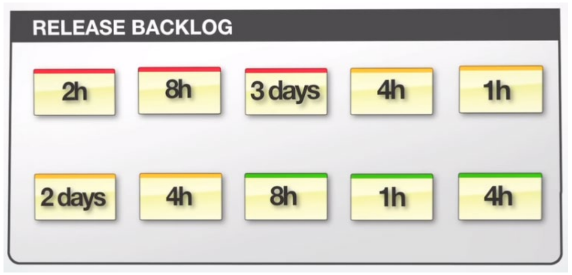
- You will get *feature* requests from a variety of stakeholders
  - Backlog: A collection of user stories (or wish list)
  - ***Must decide which goes into a release***

Release planning

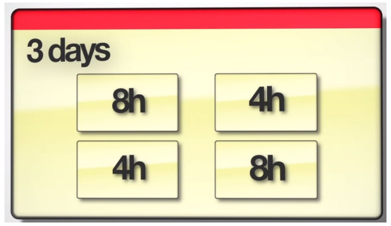
- Start with the user stories and create a release backlog



- Prioritize and estimate time for each

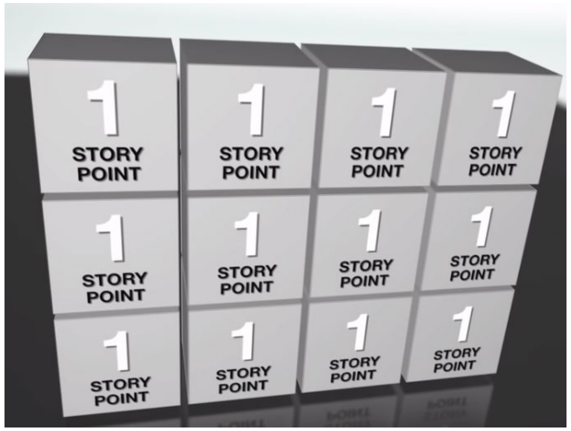


- Larger one may be broken down

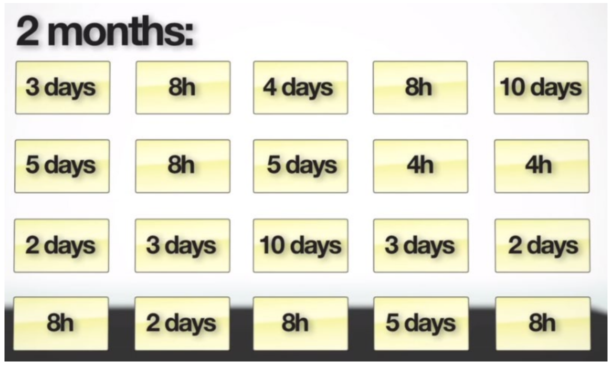


How to estimate?

- Story points?



- Or in terms of hours, days, months

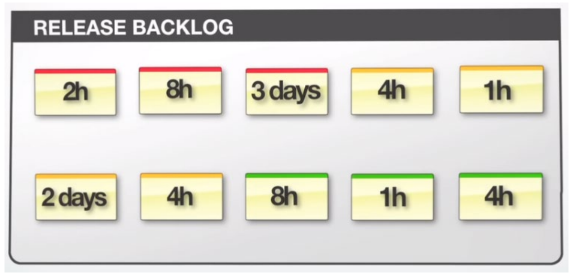


2 months:

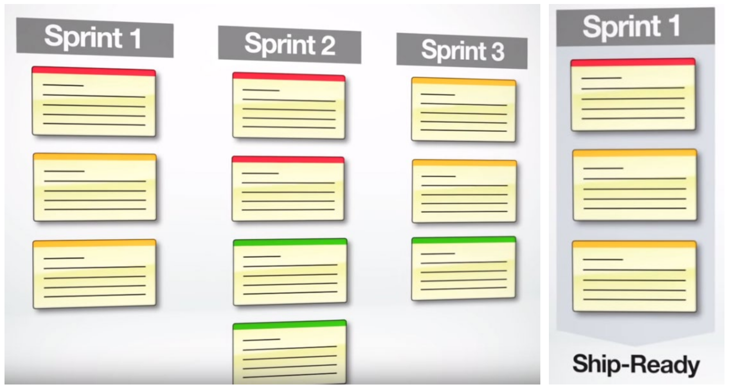
3 days	8h	4 days	8h	10 days
5 days	8h	5 days	4h	4h
2 days	3 days	10 days	3 days	2 days
8h	2 days	8h	5 days	8h

Sprints

- Start with the release backlog

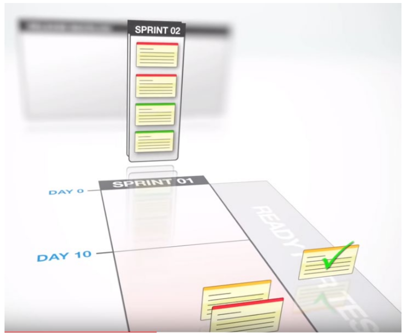


- May need several sprints to get the work done
  - Sprints: short duration milestones
  - 2-30 days (depending on release cycles)

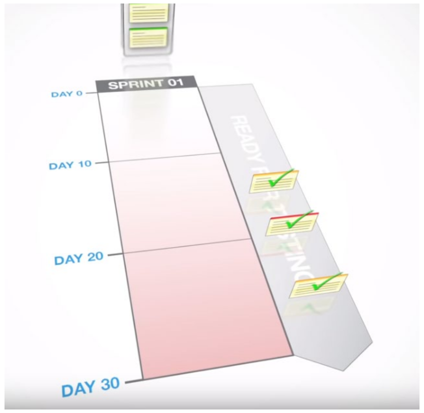


Sprint goal

- To get a subset of a release backlog to a ship-ready state

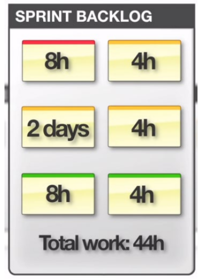


- At the end of each sprint, you have a fully tested product with all features

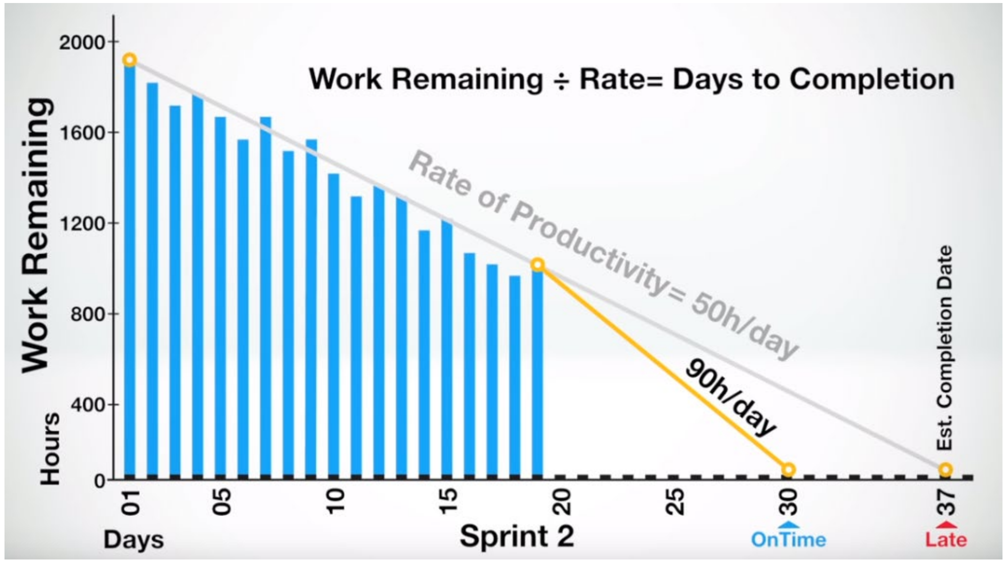


Burndown chart

- Use data comes from the release backlog to create a burnout chart

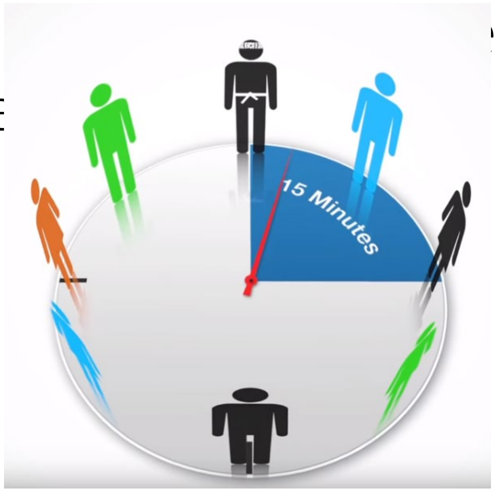


- A burndown chart is a visibility tool to ensure a project is progressing smoothly
- Shows day-by-day amount of work to be done for a given sprint



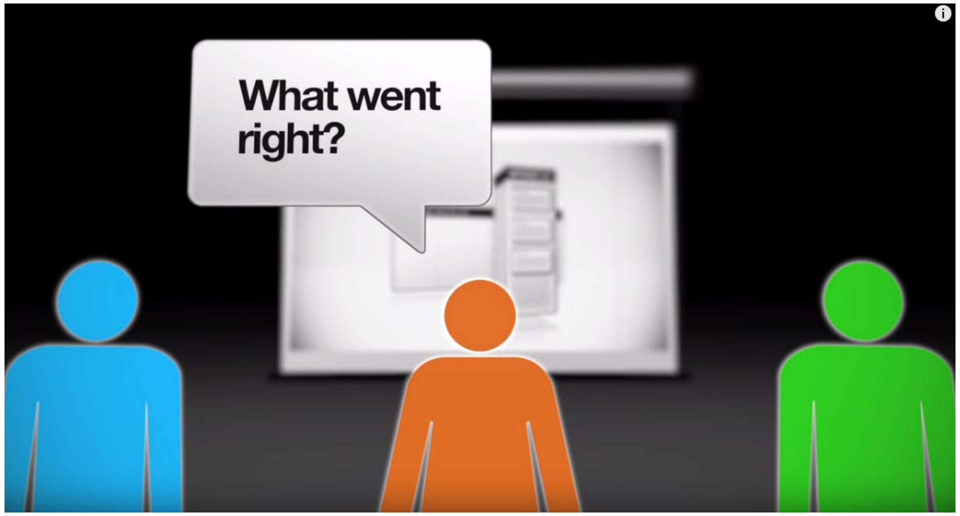
Daily scrum

- A daily tool to facilitate free flow of information between the team members
  - Stand up team member meetings
  - Team members list completed, any obstacles they obse



Sprint retrospective meeting

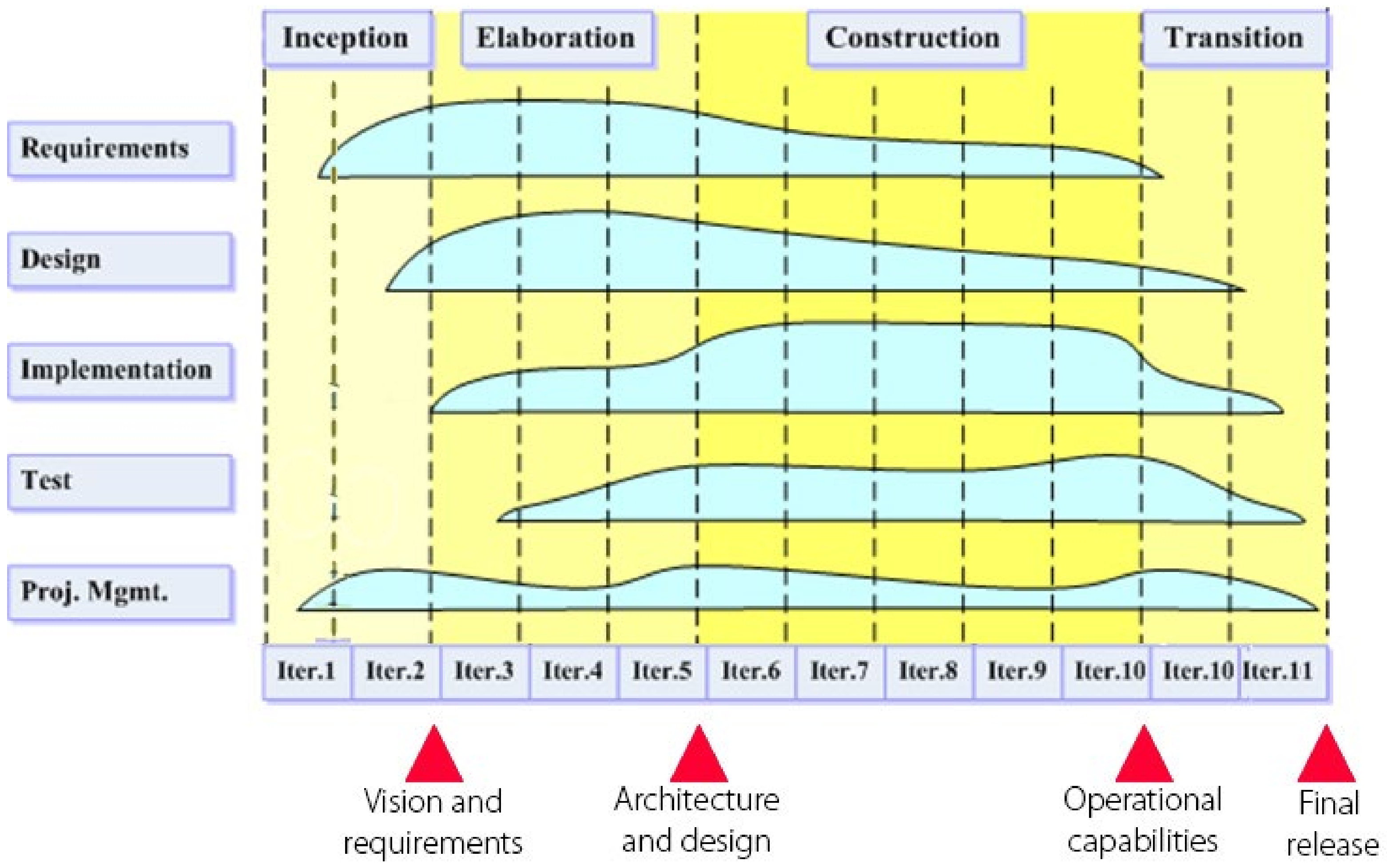
- At the end of each sprint, it is important to have a retrospective about what went right (to repeat) and what went bad (to avoid)



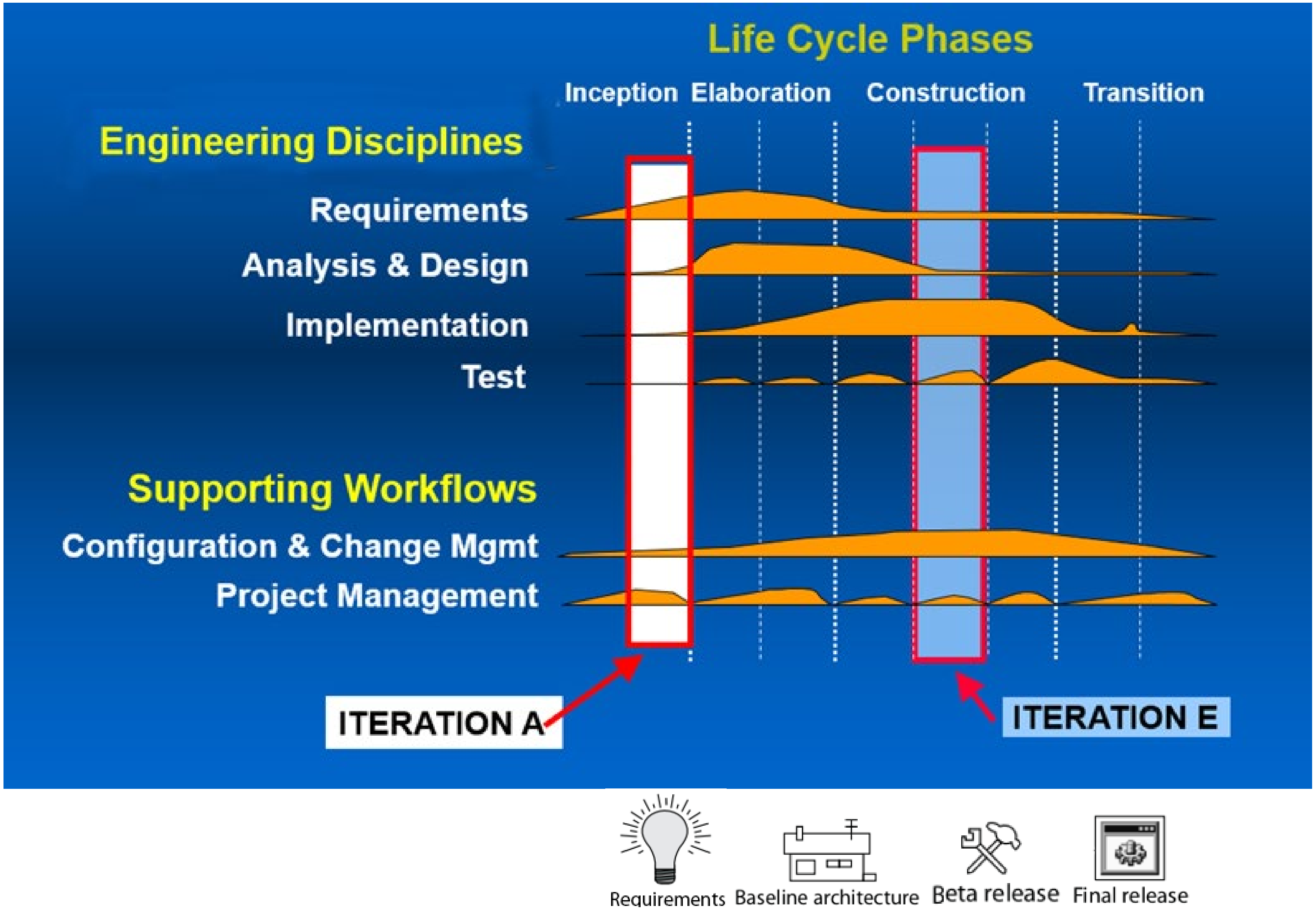
The Unified Process

- The Unified Process (UP) is an industry standard software engineering process
- IBM
- Iterative, incremental
- Four major phases (each phase has a milestone)
  - Inception (vision, domain model; requirements)
  - Elaboration (software architecture)
  - Construction (initial operational capacity)
  - Transition (final release)

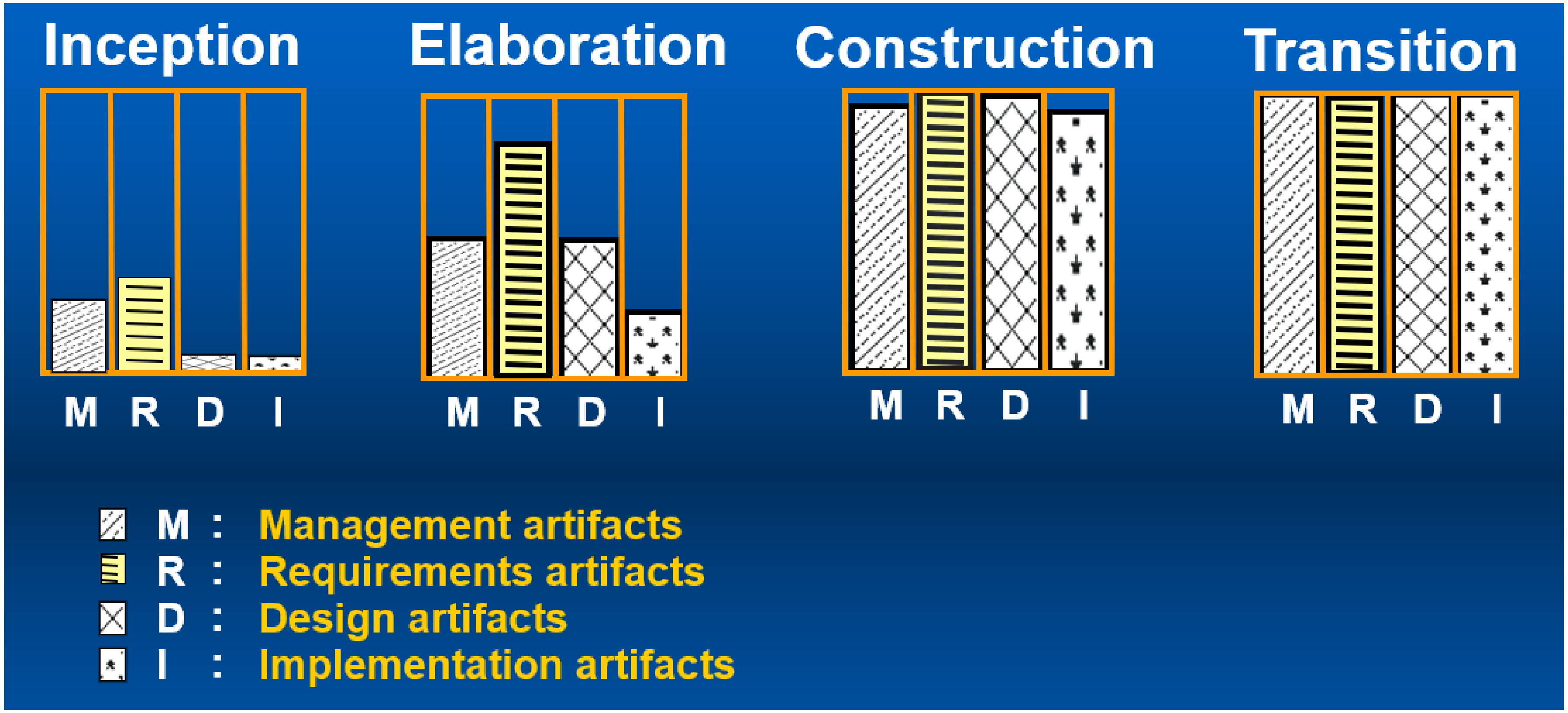
UP: A pictorial representation



The Unified Process



UP: A pictorial representation



Summary

- Four major activities (very broad)
  - Requirements engineering
  - Design
  - Coding/implementation
  - Testing
- Many software development models
  - Agile
  - Planned, disciplined approached (waterfall)
  - Incremental/iterative models