UNIT - 5

ROLES AND RESPONSIBILITIES

TEAM AROUND THE PROJECT

Designer
Backend
Business Owner
Devops
Product Manager
Frontend
Quality Assurance

BUSINESS OWNER

- Make decisions for the company.
- Responsible for identifying new businesses and partnership.
- Work at an abstract business level.
  - Defining the problem
  - How it can be tackled
  - Measure the success
- Negotiate terms of partnership
- Keeps track of business metrics to improve market and respond to business changes.
- Establish prices of product.
  - Involved in B2B (Business to Business) Communication

PRODUCT MANAGERS

- Prioritization of tasks
- Support nontechnical teams (e.g. marketing)
- Translating ideas and concepts into products and features.
- Need both soft skills and technical skills.
- Responsible for writing user stories
- Supply acceptance criteria.
- Preparing a roadmap of the project and meeting user needs.

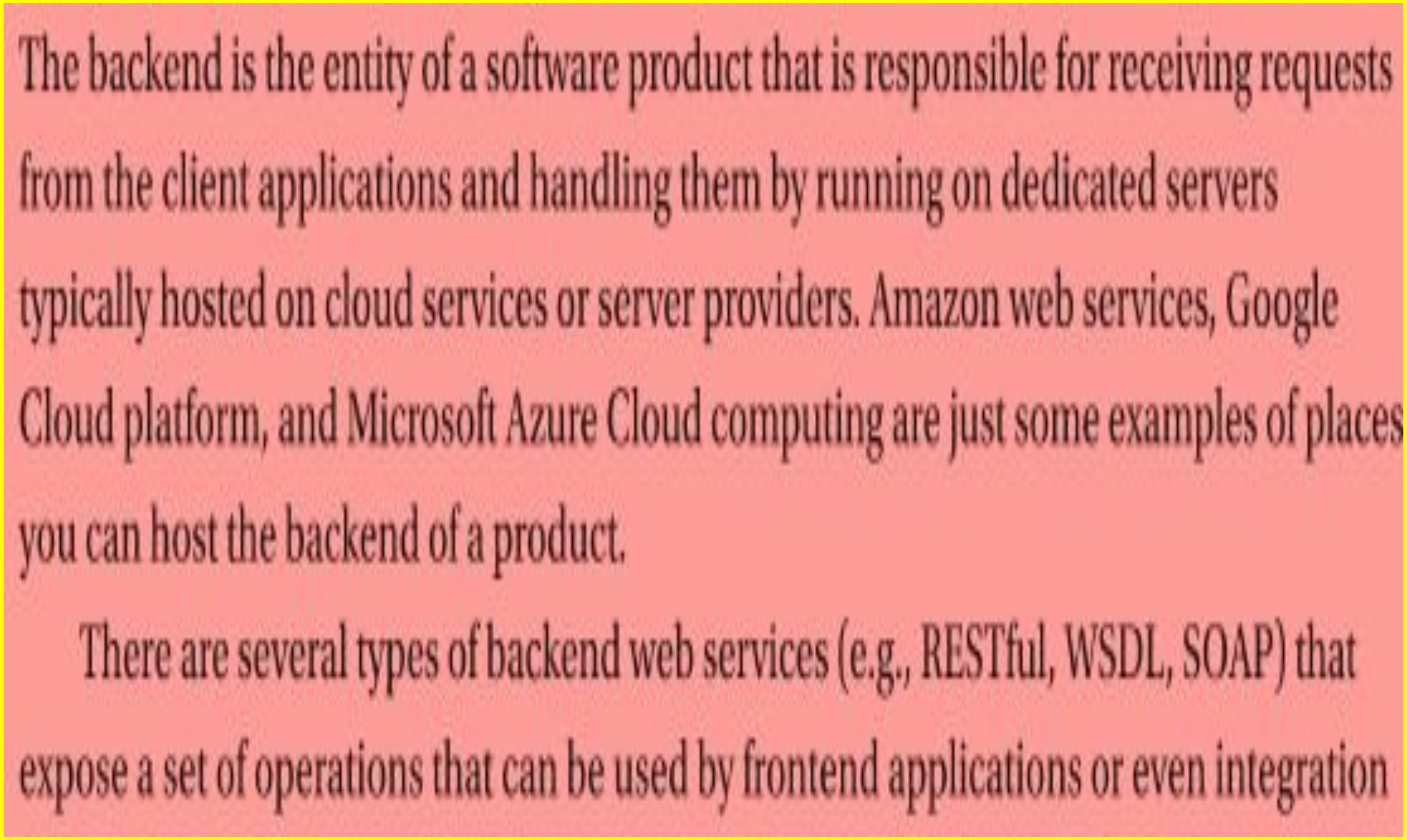
DESIGNERS

- Create interfaces
- Bring identity to the product.
- Bring consistency to the product across platforms.
- Understand how users interact with products.
- Work closely with developers to assess viability of implementation.
- Must accept challenges for maintaining consistency across operating systems.

Backend

- Entity of a software process that receives requests from client applications and handling them by running on dedicates services hosted by cloud services or service providers.
- Focus on exposing operations
- Backend Operations challenged during maintenance phase

BACKEND



The backend is the entity of a software product that is responsible for receiving requests from the client applications and handling them by running on dedicated servers typically hosted on cloud services or server providers. Amazon web services, Google Cloud platform, and Microsoft Azure Cloud computing are just some examples of places you can host the backend of a product.

There are several types of backend web services (e.g., RESTful, WSDL, SOAP) that expose a set of operations that can be used by frontend applications or even integration

BACKEND(Contd.) CHALLENGES

- Performance aspects
- Authentication
- Authorisation
- Validation

FRONTEND

- Visible to the enduser.
- Can retrieve, store, modify and delete data entities of an application.
- Piece of software that runs close to client side.
- Two modules – Representation and Logic
  - Representation – what user sees, interface, how elements are rendered and interaction with them.
  - Logic – Fetching, Handling Requests, States, Validation of Data Input.
- In some companies, frontend implementation may start even before design is complete.
- Responsible to implement interfaces and logic of application that interacts with users.

QUALITY ASSURANCE

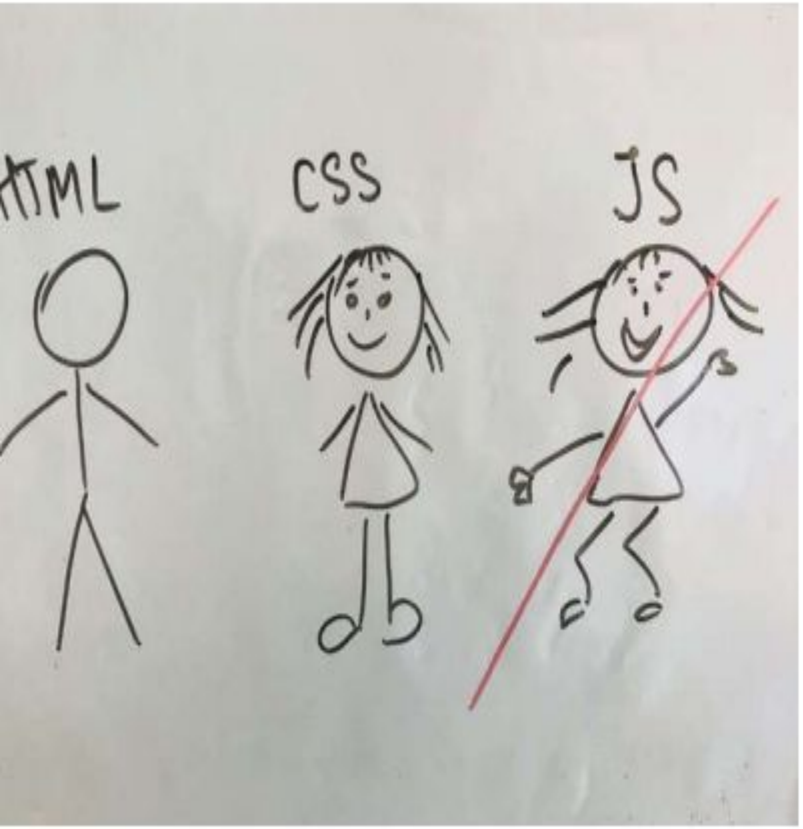
- Responsible for making sure that everything that goes to the enduser meets the requirements.
- Structure of QA departments depend on the size of the company they are working
- QA teams perform several tests to ensure that everything is implemented according to acceptance criteria.
- QA teams also focus on regression testing.
- QA teams also focus on system testing (check that system works in all environments) (stress & performance testing)
- QA teams can pick up issues with a feature (which may lead to feature rejection or rework (iterated from start)

DevOps

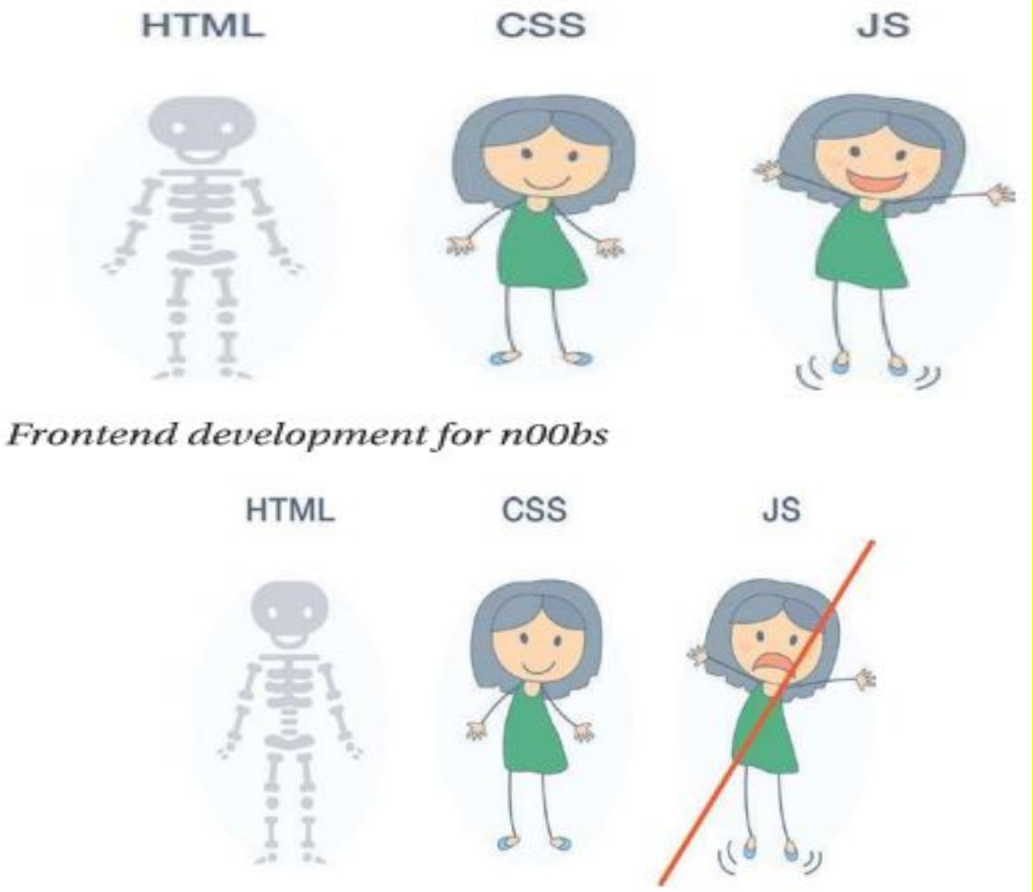
- Development + Operations
- Responsible for operational steps of the development and infrastructure.
- Responsible for building continuous integration and delivery pipelines, managing the servers, performing migrations and doing actual deployments.
- Responsible for making sure current infrastructure can handle expected load and still performs.

- Concept of User Interface(UI) and User Experience (UX)
  - Differences : Explanation
- User Interface engineering – focused on defining a clear structure and dressing up nicely to interest a target audience.

DIAGRAMATIC REPRESENTATION



Whiteboard sketch explaining UI engineering



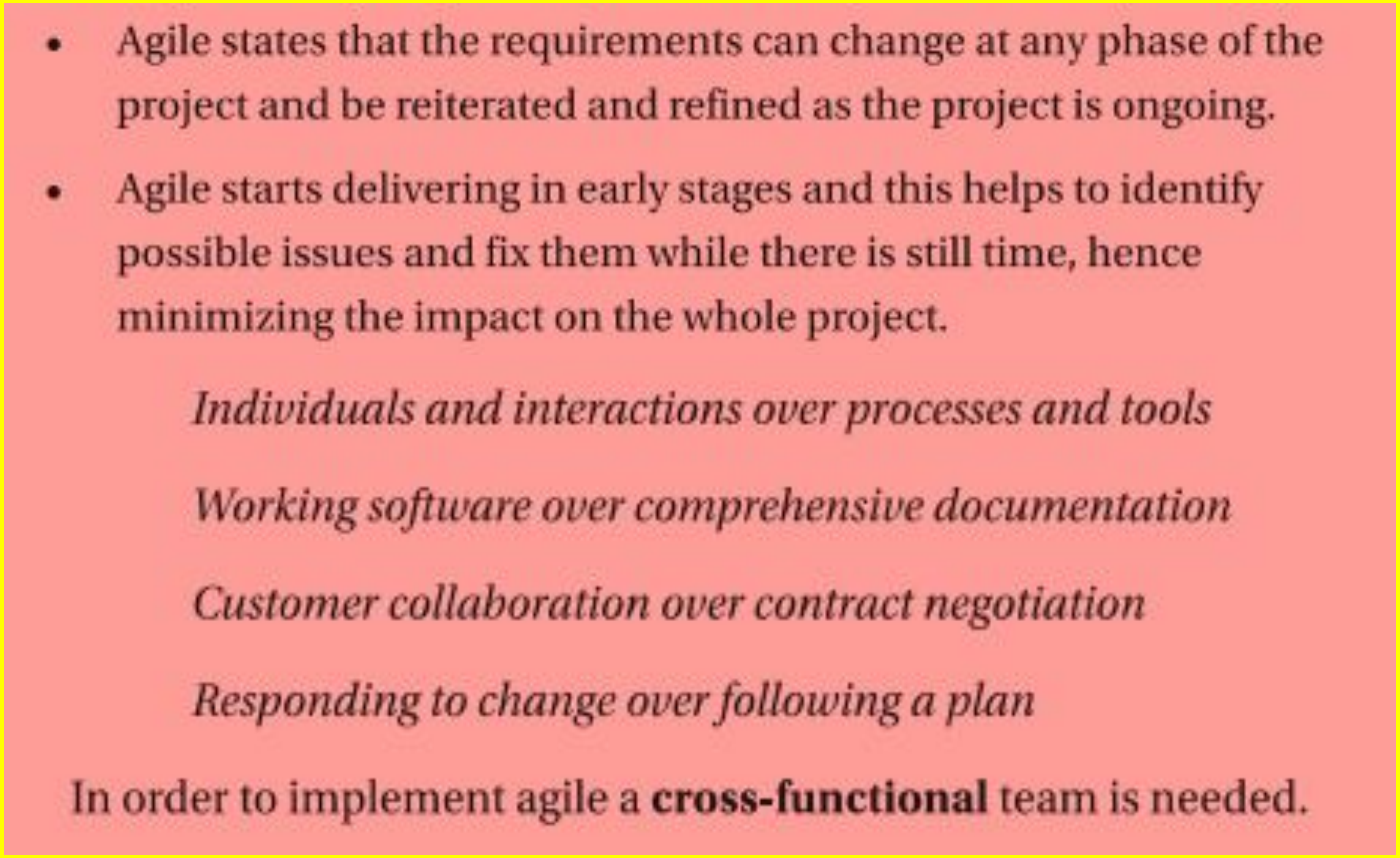
UI engineering does not include the dynamic part of the frontend development

DRAWBACKS OF WATERFALL MODEL

PRINCIPLES OF AGILE METHODOLOGY

- Everything always changes
- Deal with frequent software deliveries

DIFFERENCES WITH WATERFALL MODEL



- Agile states that the requirements can change at any phase of the project and be reiterated and refined as the project is ongoing.
- Agile starts delivering in early stages and this helps to identify possible issues and fix them while there is still time, hence minimizing the impact on the whole project.

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

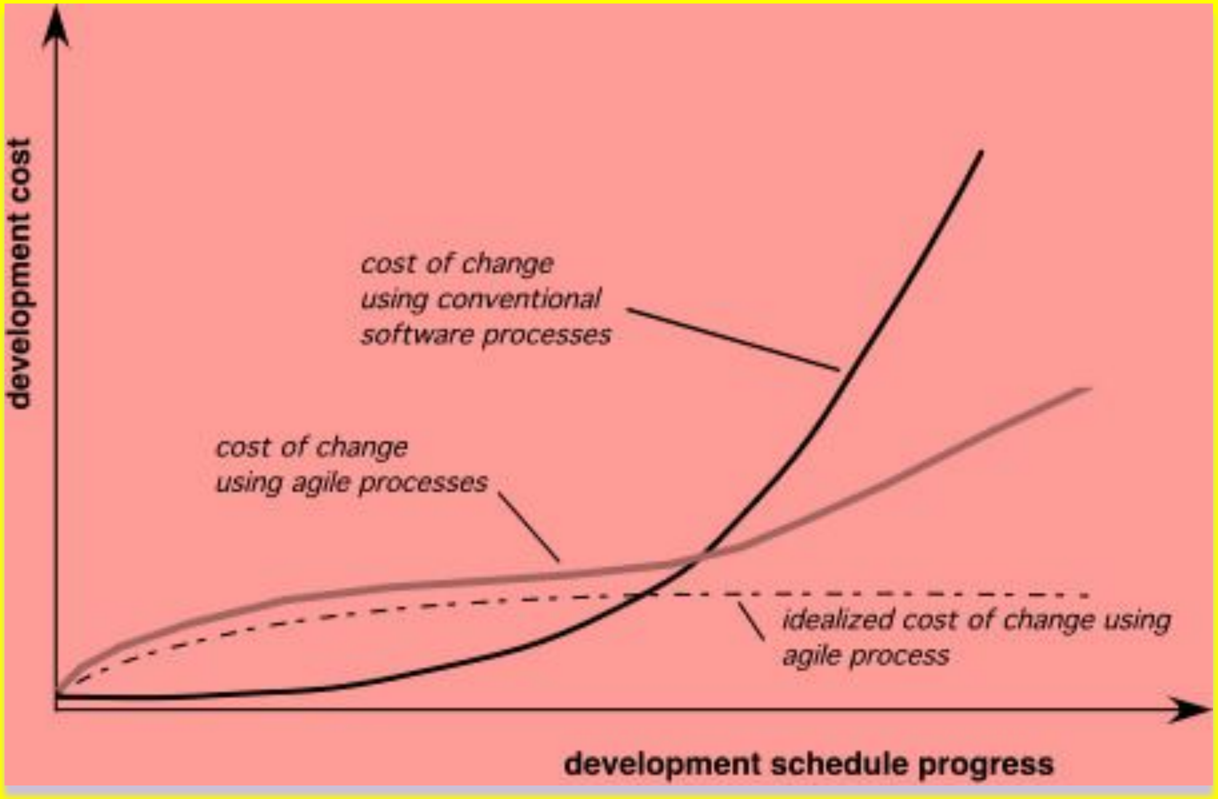
Responding to change over following a plan

In order to implement agile a **cross-functional** team is needed.

What is “Agility”?

- Effective (rapid and adaptive) response to change
- Effective communication among all stakeholders
- Drawing the customer onto the team
- Organizing a team so that it is in control of the work performed
- Rapid, incremental delivery of software

Agility and the Cost of Change



An Agile Process

- Is driven by customer descriptions of what is required (scenarios)
- Recognizes that plans are short-lived
- Develops software iteratively with a heavy emphasis on construction activities
- Delivers multiple ‘software increments’
- Adapts as changes occur

Agility Principles

- Highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development.
- Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face–to–face conversation.

Agility Principles (Contd.)

- Working software is the primary measure of progress.
- Agile processes promote sustainable development.
  - The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity – the art of maximizing the amount of work not done – is essential.
- The best architectures, requirements, and designs emerge from self–organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Human Factors

- Process molds to the needs of the people and team, not the other way around
- Key traits must exist among the people on an agile team and the team itself:
  - **Competence.**
  - **Common focus.**
  - **Collaboration.**
  - **Decision-making ability.**
  - **Fuzzy problem-solving ability.**
  - **Mutual trust and respect.**
  - **Self-organization.**

Extreme Programming (XP)

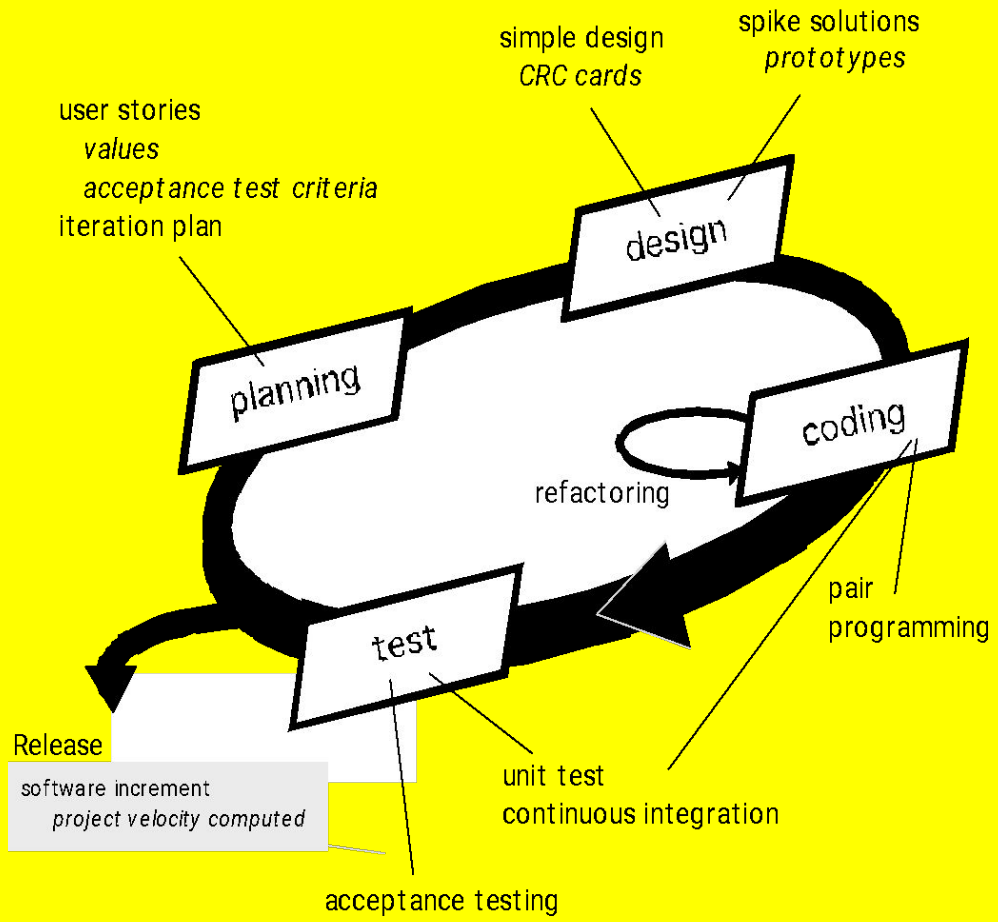
- Most widely used agile process, originally proposed by Kent Beck
- XP Planning
  - Begins with the creation of "user stories"
  - Agile team assesses each story and assigns a cost
  - Stories are grouped to for a deliverable increment
  - A commitment is made on delivery date
  - After the first increment "project velocity" is used to help define subsequent delivery dates for other increments

Extreme Programming (XP) (Contd.)

- XP Design
  - Follows KIS principle
  - Encourage the use of CRC cards
  - For difficult design problems, suggests the creation of “spike solutions”—a design prototype
  - Encourages “refactoring”—an iterative refinement of the internal program design

- XP Coding
  - Recommends the construction of a unit test for a store before coding commences
  - Encourages "pair programming"
- XP Testing
  - All unit tests are executed daily
  - "Acceptance tests" are defined by the customer and executed to assess customer visible functionality

Extreme Programming (XP) (Contd.)



XP PRACTICES

- Sustainable pace
- Collective code ownership
- Test-driven development
- Continuous integration
- Coding standards
- Refactoring

Agile Modeling

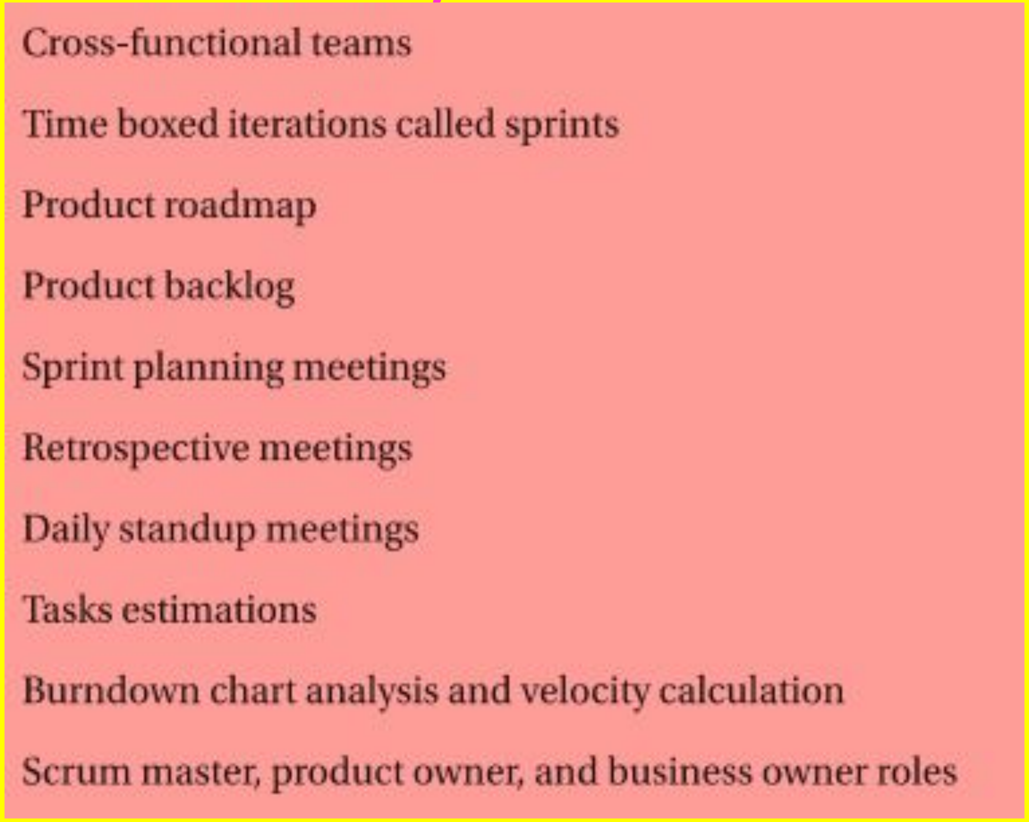
- Originally proposed by Scott Ambler
- Suggests a set of agile modeling principles
  - Model with a purpose
  - Use multiple models
  - Travel light
  - Content is more important than representation
  - Know the models and the tools you use to create them
  - Adapt locally

Scrum

- Originally proposed by Schwaber and Beedle
- Framework that implements agile patterns
- Scrum—distinguishing features
  - Development work is partitioned into “packets”
  - Testing and documentation are on-going as the product is constructed
  - Work occurs in “sprints” and is derived from a “backlog” of existing requirements
  - Meetings are very short and sometimes conducted without chairs
  - “demos” are delivered to the customer with the time-box allocated

SCRUM

- Prescribes well defined roles and strict rules.
- Principles of SCRUM rely on



Cross-functional teams

Time boxed iterations called sprints

Product roadmap

Product backlog

Sprint planning meetings

Retrospective meetings

Daily standup meetings

Tasks estimations

Burndown chart analysis and velocity calculation

Scrum master, product owner, and business owner roles

WORK OF SCRUM

- Start from business and priority discussion
- Discussion between Business Owner, Product owner and team – High level description
  - Product owner splits into small chunks.
- Work that needs to be done – put in backlog
  - Sprint backlog
    - Estimated list of features to be worked on the next iteration.
  - Product backlog
    - Prioritized set of features to be developed or bugs to be corrected

WORK OF SCRUM(Contd.)

- Features described from user’s perspective – User stories
- Product owner has to plan and prioritize backlog according to business needs and goals.

SPRINTS

- Chunks of work to be distributed along timebox iterations.
- Duration of sprint – 2 to 4 weeks
- Planning Meeting + retrospective meeting
- Planned in the beginning, Analyzed in the end.

SCRUMMASTER

- Effective communicator
- Build trust among members (stakeholders)
- Can evolutnize development

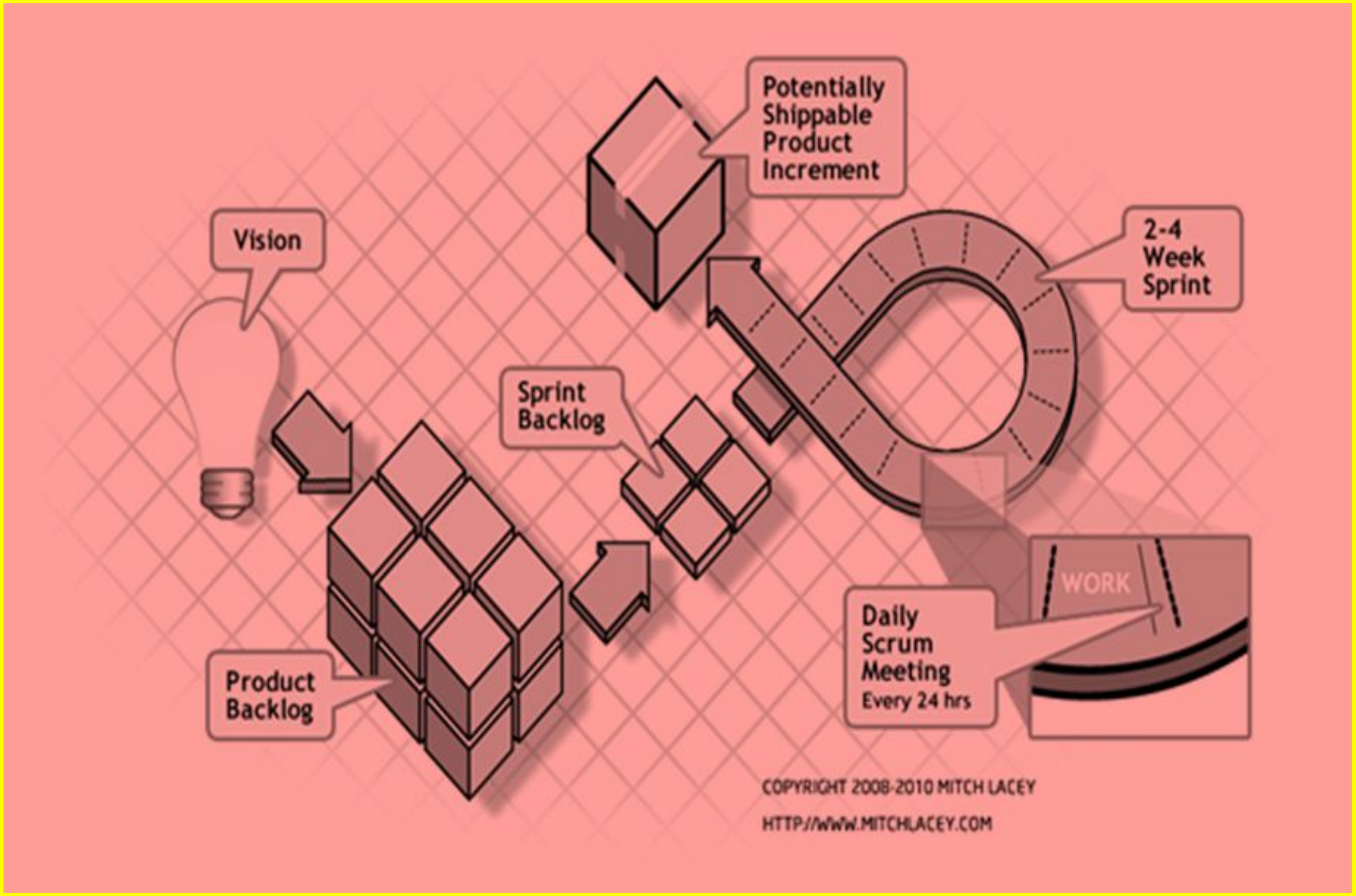
PRODUCT OWNER

- Working with the stakeholders to understand the functionality of the system under development
- Write user stories or work jointly with the customers to write them.
  - The stories go into the Product Backlog
- Manages and represents the interests of the stakeholders and customers.
- Responsible for the success or failure of the project.

SCRUM TEAM

- The core team executes the Product Owner's vision with the help of the ScrumMaster.
- The team is comprised of the people needed to deliver the work –developers, testers, architects, designers – anyone who is needed.
  - Concept of Cross Functionality
- A core team is ideally made up of full-time people dedicated to the project.
- The team is responsible for managing its work, its commitments and the execution of those commitments

- RETROSPECTIVE MEETING :
  - Scrum Master analyzes burndown chart, calculates team velocity and team discusses what went well during sprint and what problems faced
- SPRINT PLANNING MEETING
  - During the meeting, the team analyzes the tasks that makes into the sprint, estimates and commits to them.


Potentially Shippable Product Increment
Vision
2-4 Week Sprint
Sprint Backlog
WORK
Daily Scrum Meeting Every 24 hrs
Product Backlog
COPYRIGHT 2008-2010 MITCH LACEY
HTTP://WWW.MITCHLACEY.COM

SPRINT BACKLOG

- Output of the Sprint Planning meeting
- Essentially the list of tasks that the Scrum team needs to complete during the sprint in order to turn a selected set of product backlog items into a deliverable increment of functionality
- Have a time-based (hourly) estimate
- Scrum Team adds and removes items from Sprint Backlog

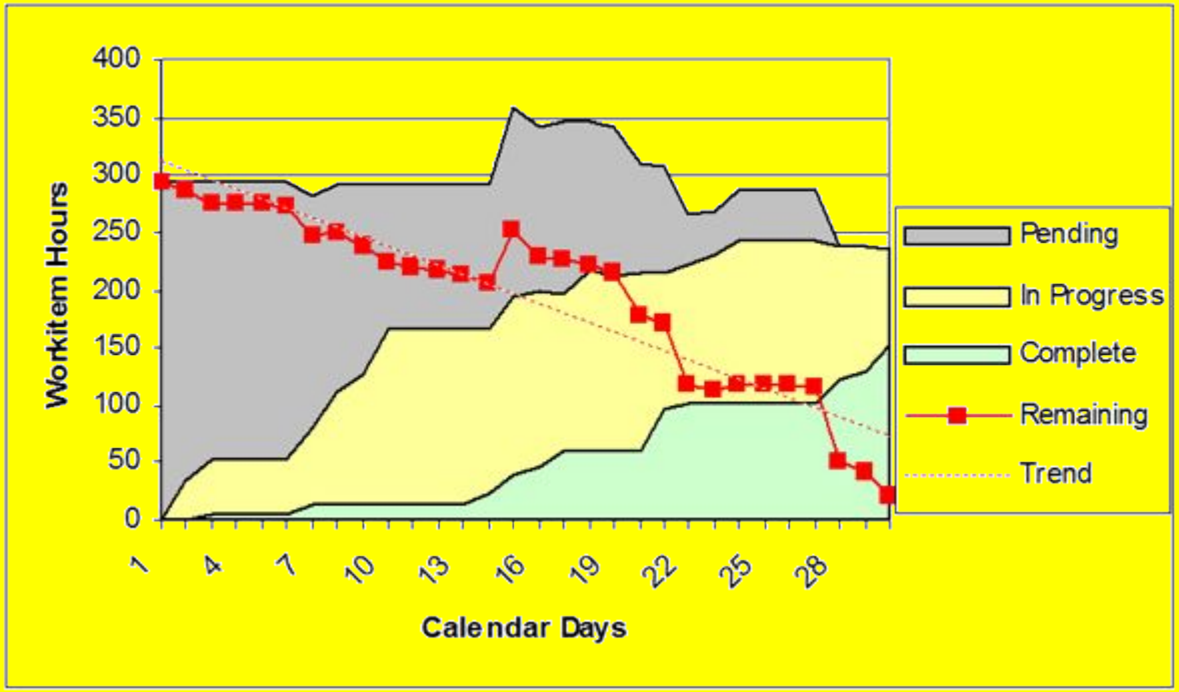
PRODUCT BACKLOG

- Is a prioritized, ordered list, sorted by business value and risk.
- Contains the work needed to accomplish the project.
- Contains user stories but may also contain functional requirements, nonfunctional requirements, bugs and various issues.
- Estimated in abstract units such as story points, which use a relative weighting model.
- Owned and managed by the Product Owner.
- The release plan can be derived through calculations on the Product Backlog.

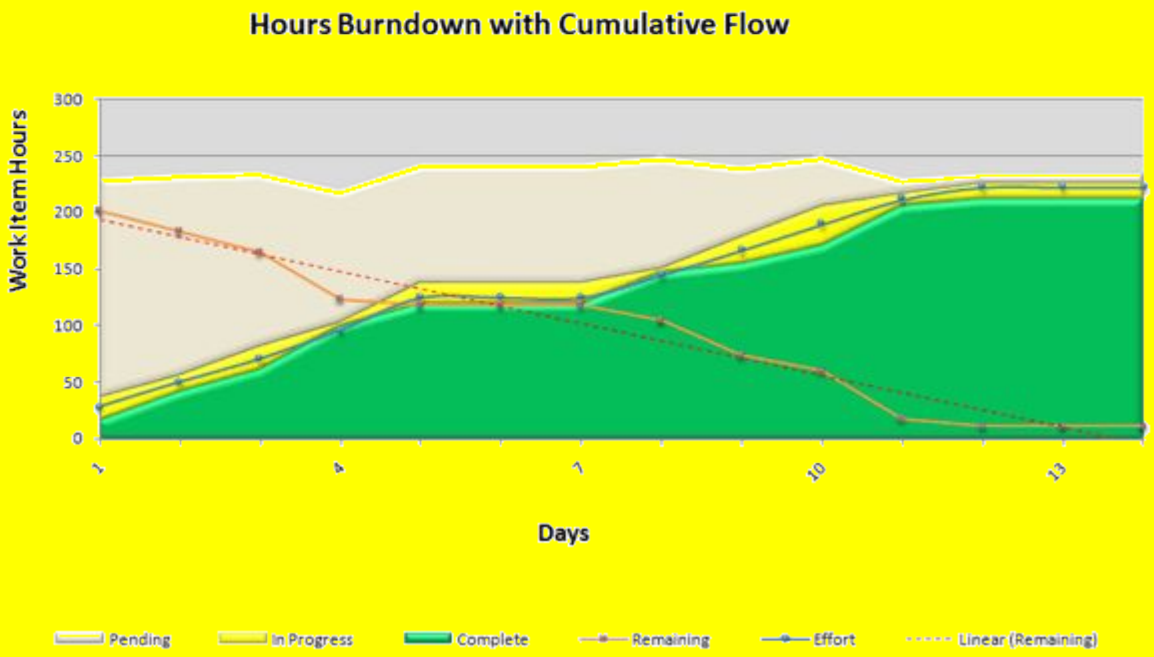
PRODUCT BACKLOG(Contd.)

- Product Owner can estimate what stories will be complete and when, by viewing the size of the story and fitting stories into Sprits based on the velocity the team is able to execute
  - Velocity – How much product backlog effort the team can handle in one sprint
- Product Backlog should follow a model (Bill Wake's INVEST model) where stories should be:
- I - Independent
- N - Negotiable
- V - Valuable
- E - Estimable
- S - Small
- T - Testable

SAMPLE BURNDOWN CHARTS



SAMPLE BURNDOWN CHARTS(Contd.)



DAILY SCRUM MEETING

- Essential to good communication.
- The meeting is time-boxed to fifteen minutes regardless of team size.
- Effective Daily Scrum meetings should be held at the same time and at the same place every day.
- Team members are required to attend.
  - If a person cannot make the meeting, they should provide status to the team prior to the meeting in a format that is agreed upon by the team.

DAILY SCRUM MEETING (Contd.)

- Each team member should respond to the three questions in Scrum:
  - What have you done since the last Daily Scrum regarding this project?
  - What will you do between now and the next Daily Scrum meeting regarding this project?
  - What impedes you from performing your work as effectively as possible?

SPRINT PLANNING MEETING

- Output of Sprint Planning meeting will be in Sprint Backlog
- Part – 1 :
  - Review of product backlog items to forecast and deliver.
  - At the end of Part -1, select a Sprint Goal
- Part – 2 :
  - Decide how the work is built
  - Decompose product backlog items into work tasks and estimate them in hours.

SPRINT REVIEW MEETING

- Part -1 : Customer Review and Demonstration
  - Show customers what is delivered during the Sprint
  - Occur on the last day of the sprint
  - One hour per week
- Part -2 : Team Retrospective
  - Occur on the last day of the sprint
  - 0.75 hours per week
- Preparation for Review
  - Maximum : One hour

CUSTOMER REVIEW

- Customers review the following Data :
  - The work the team committed to delivering
  - The work they completed
  - Key decisions that were made during the iteration/sprint (E.g. technical, market-driven, requirements)
  - Project metrics (code coverage, etc)
  - Demo of the work itself
  - Priority review (for the next iteration/sprint)

TEAM RETROSPECTIVE

- Occurs on the last day of the sprint, typically after the customer review meeting.
- The purpose of the team retrospective is to identify the things that
  - Team is doing well that they should keep doing
  - Things they should start doing in order to improve
  - Things that are keeping them from performing at their best that they should stop doing.
- The meeting is facilitated by ScrumMaster.