

Introduction to Software Development

Week 3 Day 1

Led by: Emily Crose
for
Oakland University

Previous Session Review

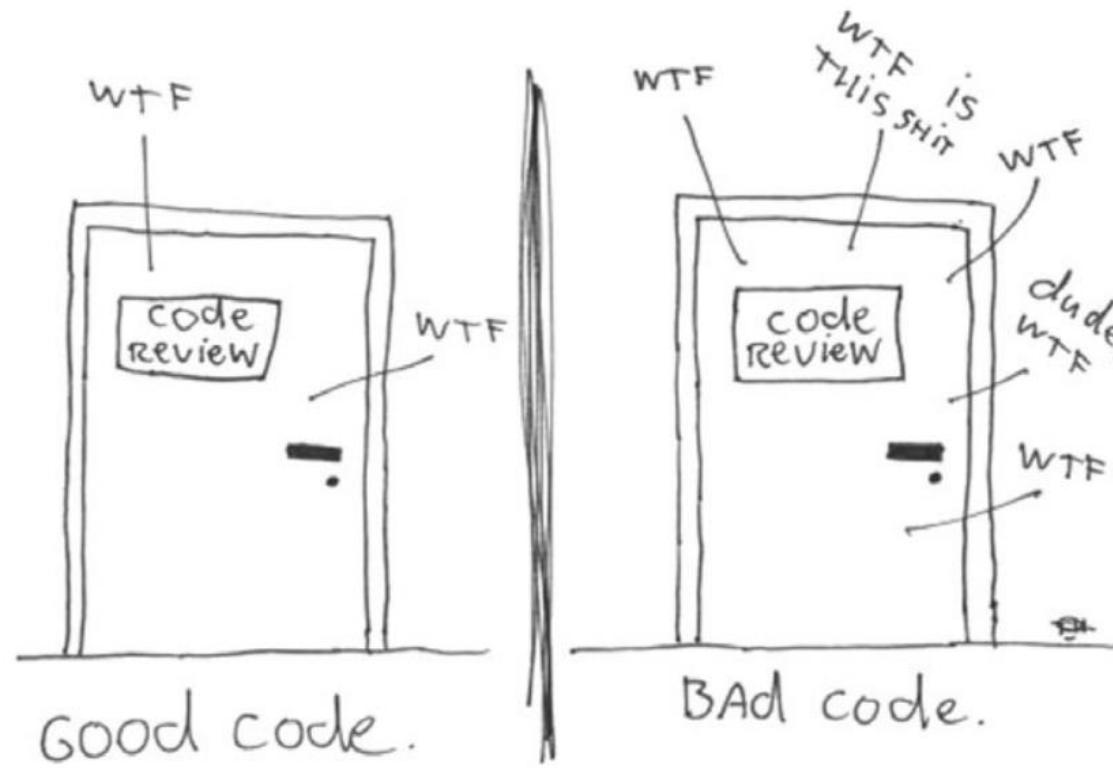
Questions From Previous Session



Terms to listen for

- Sprint
 - A period of focused development
- Roadmap
 - An overarching plan for development

The ONLY VALID MEASUREMENT OF CODE QUALITY: WTFs/minute



(c) 2008 Focus Shift

Form

Function

Minimalist
code

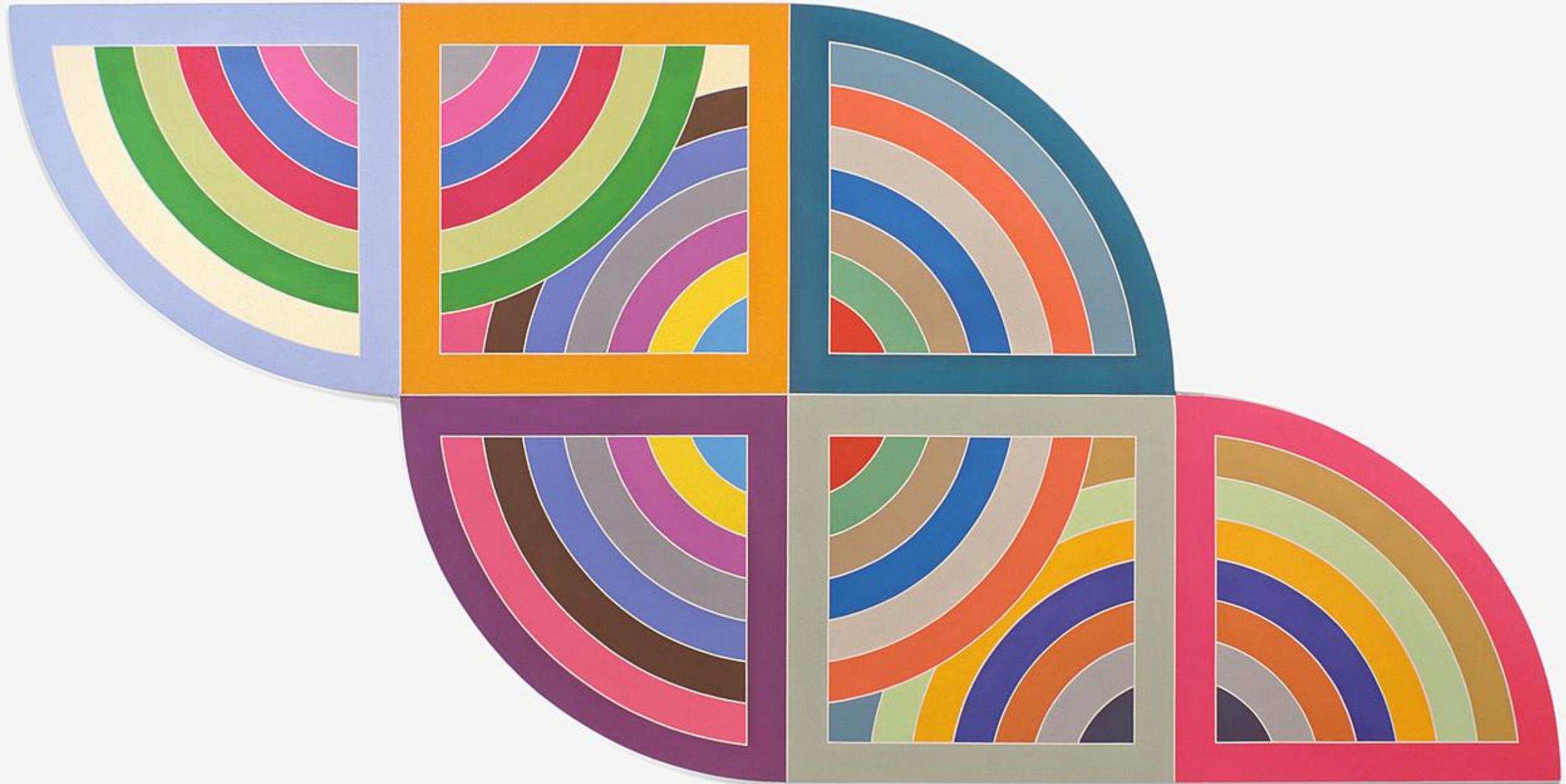
Features

Low
complexity

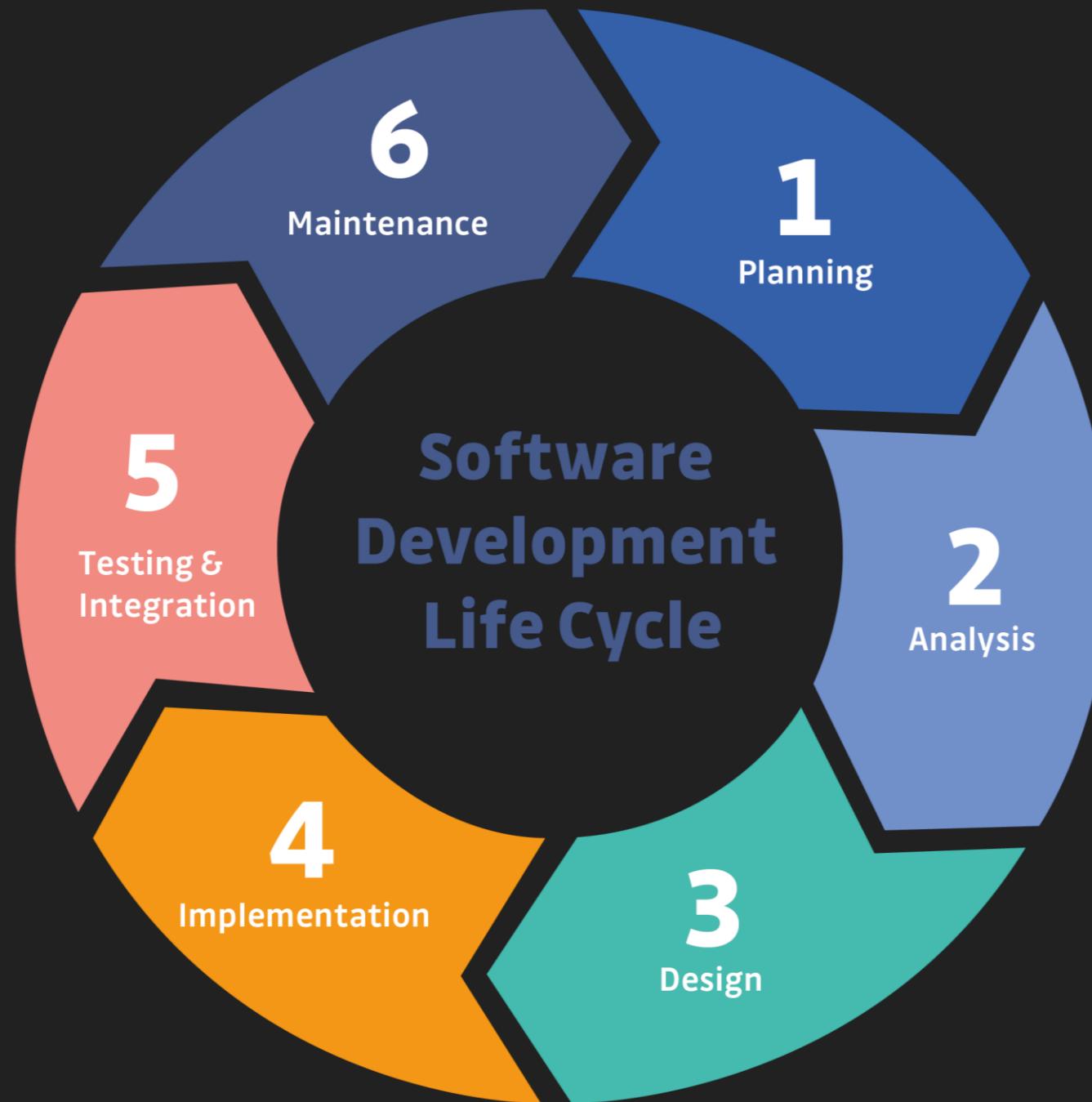
UI

Usability

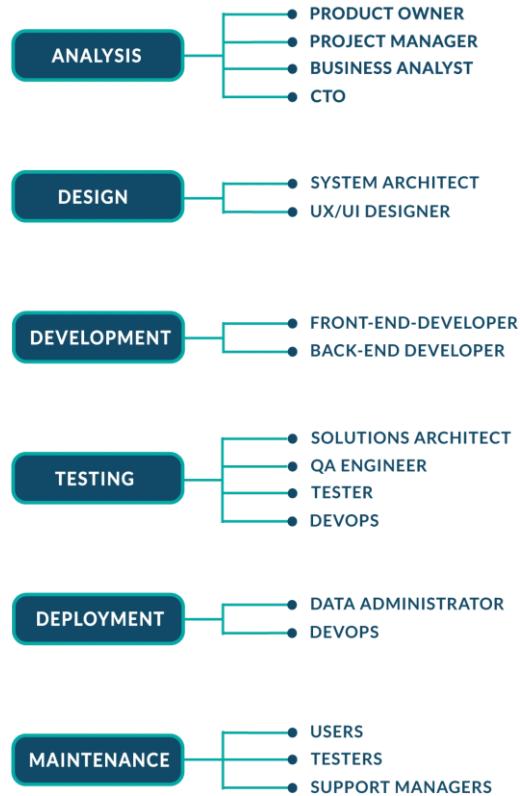


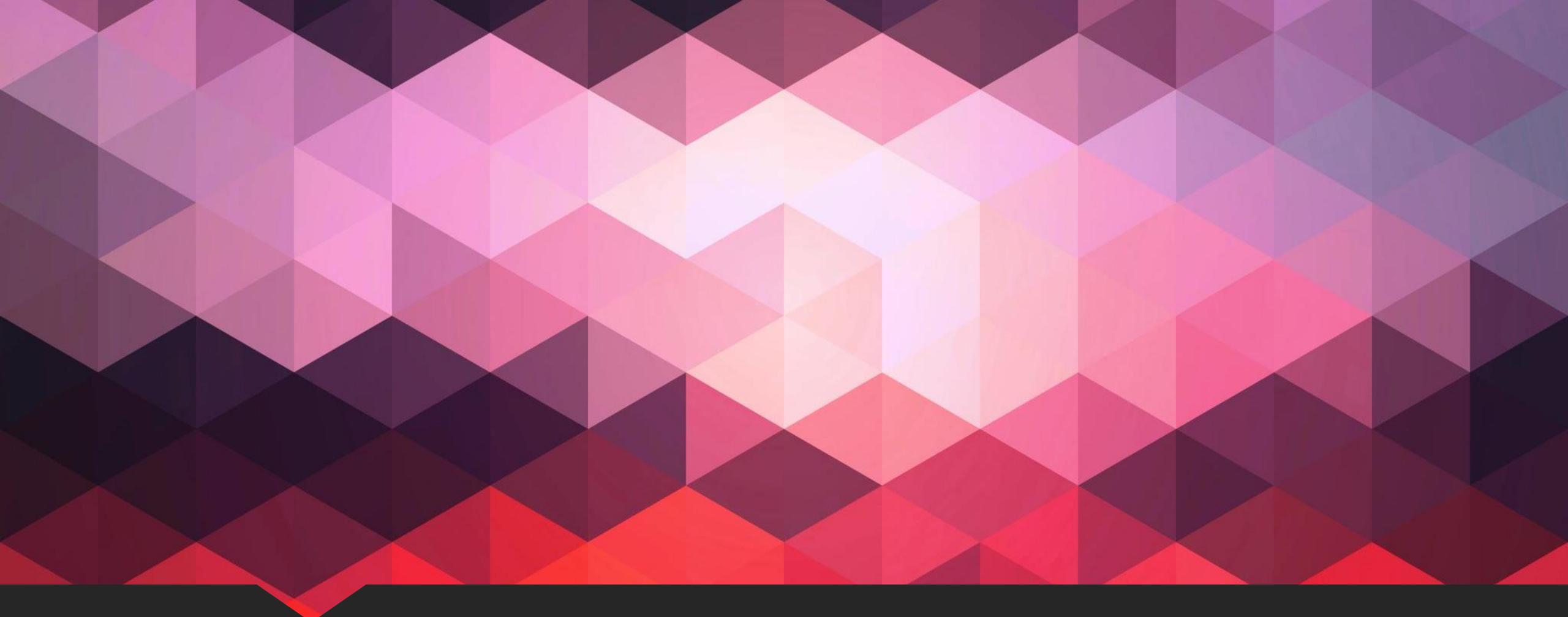


The Software Development Life Cycle



SDLC Phases





SDLC Step 1: Planning

Questions to ask ourselves in step 1

- What is our problem?
- Do we have the individuals we need to identify a solution to this problem?
- What do we need to solve the problem?
 - Is the solution a problem software can fix for us?

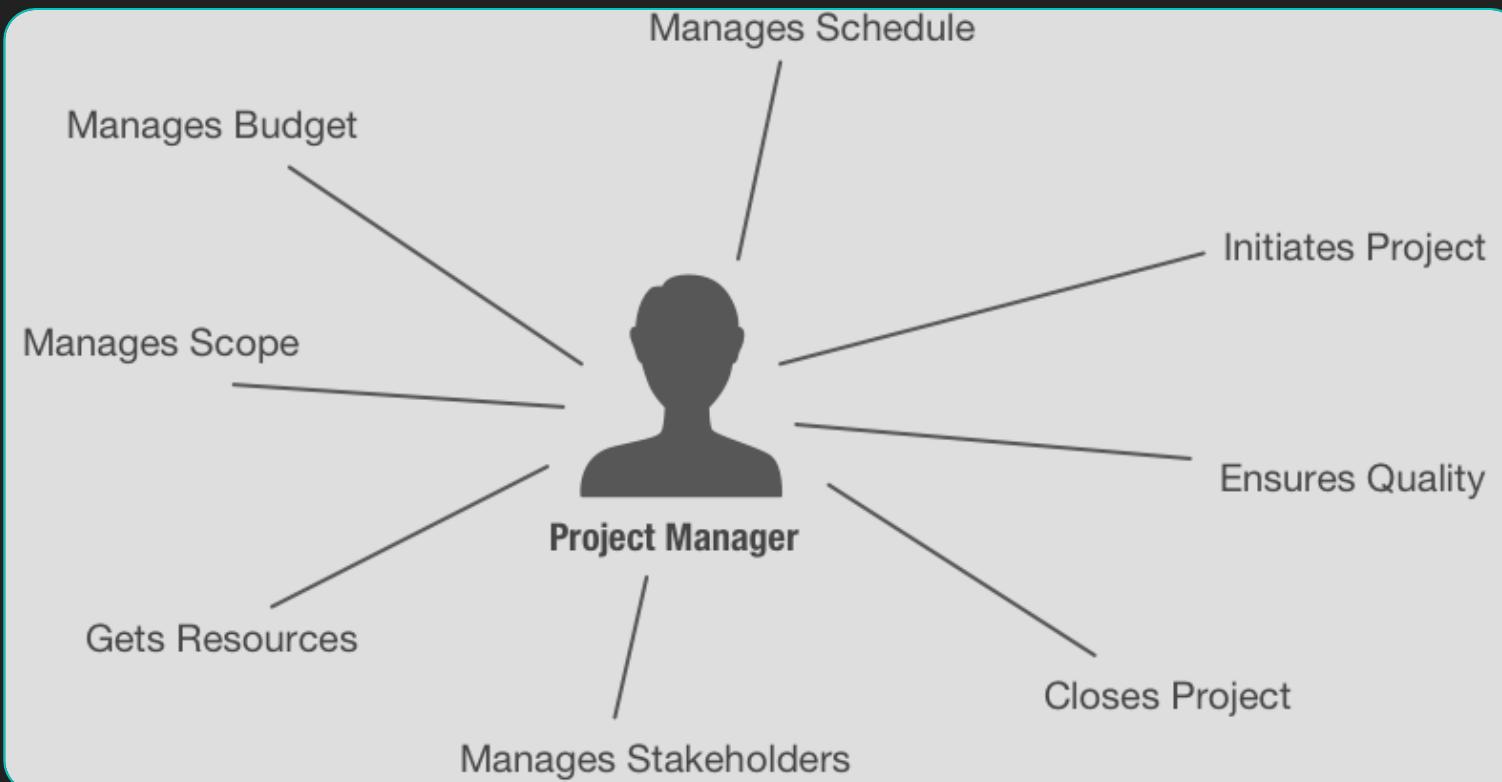
Roles in Step 1



Product Owner



Project Manager Role



Business Analyst Roles & Responsibilities

- Gathers & tracks product requirements
- Stake holder consent
- Documentation
- GAP analysis
- User testing

CTO

- Provides top-cover for decision making
- Budgeting Approval
- Stakeholder, and final approval pre & post project

Planning Activities

Gathering requirements

Planning solutions

Planning sprints

Road-mapping

Gathering Requirements

- What problems do we need to solve with an app?
- How do we our app to solve those problems?
- What environment(s) will our app run on?

Planning Solutions

- Given our requirements, what features do we need?
- What dependencies do we have to make the features functional?
 - Libraries?
- What should our interface look like?

Tasking

- “Tactical” level jobs
- Should be single-focus
- Do NOT overload tasks!
- Can include bug-fix, but should be kept separate

Planning Sprints

- Collection of tasks
- Planned around Mid-Term Goals
- Typically planned for a month or more of focused worktime
- Helps stay organized and focused

Sprints Cont'

- Must be accomplishable!
- Organized around quarterly goals
- Only ONE sprint per team

Road-Mapping

- Long-term goals
- Plans multiple sprints in advance



10 minute
break

Tools To Manage Planning

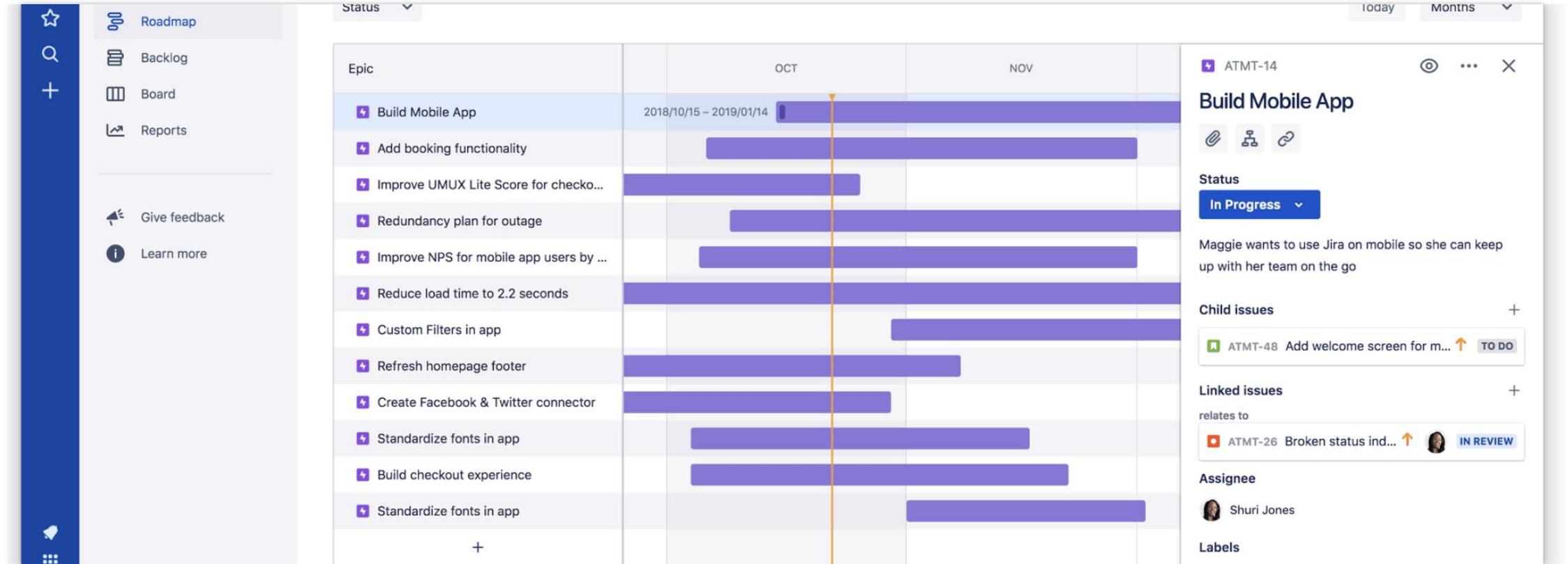


Organizational Tool: Jira

Jira Features

- Create & assign tasks (or “tickets”)
- Track ticket progress
- Organize tasks
- Integrate with other Atlassian organizational tools
 - Confluence
 - Bitbucket

Road-Mapping Tools



Epics

Overall tasks
are organized
into Epics

Epics are
made up of
tasks

Each task can
have subtasks

Step 2: Analysis

Questions to ask in step 2

- Does a software solution already exist for this problem?
 - If yes:
 - Can we legally/contractually use it?
 - How much will it cost for us to use it?
- Do we need to create our own solution?
 - How much will it cost for us to build it?
 - Can we afford this cost?
 - How much will it cost for us to support this product?
 - Are we committed to supporting this new product long-term?

Step 2: Analysis Roles



Analysis Activities

Cost/Benefit Analysis

- This project will cost \$X, what can we expect to see in return for this cost?
- Is the value tangible or intangible?
 - Is the intangible benefit worth the cost?
- Can we realistically afford to do this year-over-year
- How much do we believe this product will cost to maintain over 3 years? 5 years?
- Do we currently have the resources we need to support this project, or do we need to hire people?
 - If we don't have enough devs, how many people will we need to hire?

Opportunity Cost Analysis

- If we spend our time on this project, what are we NOT doing with our resources?
- Do we have our priorities in order?

Agile Development Model

- Work iteratively
- Integrates feedback from customers
- Creates achievable goals at all stages
- Reduce feature engineering complexity

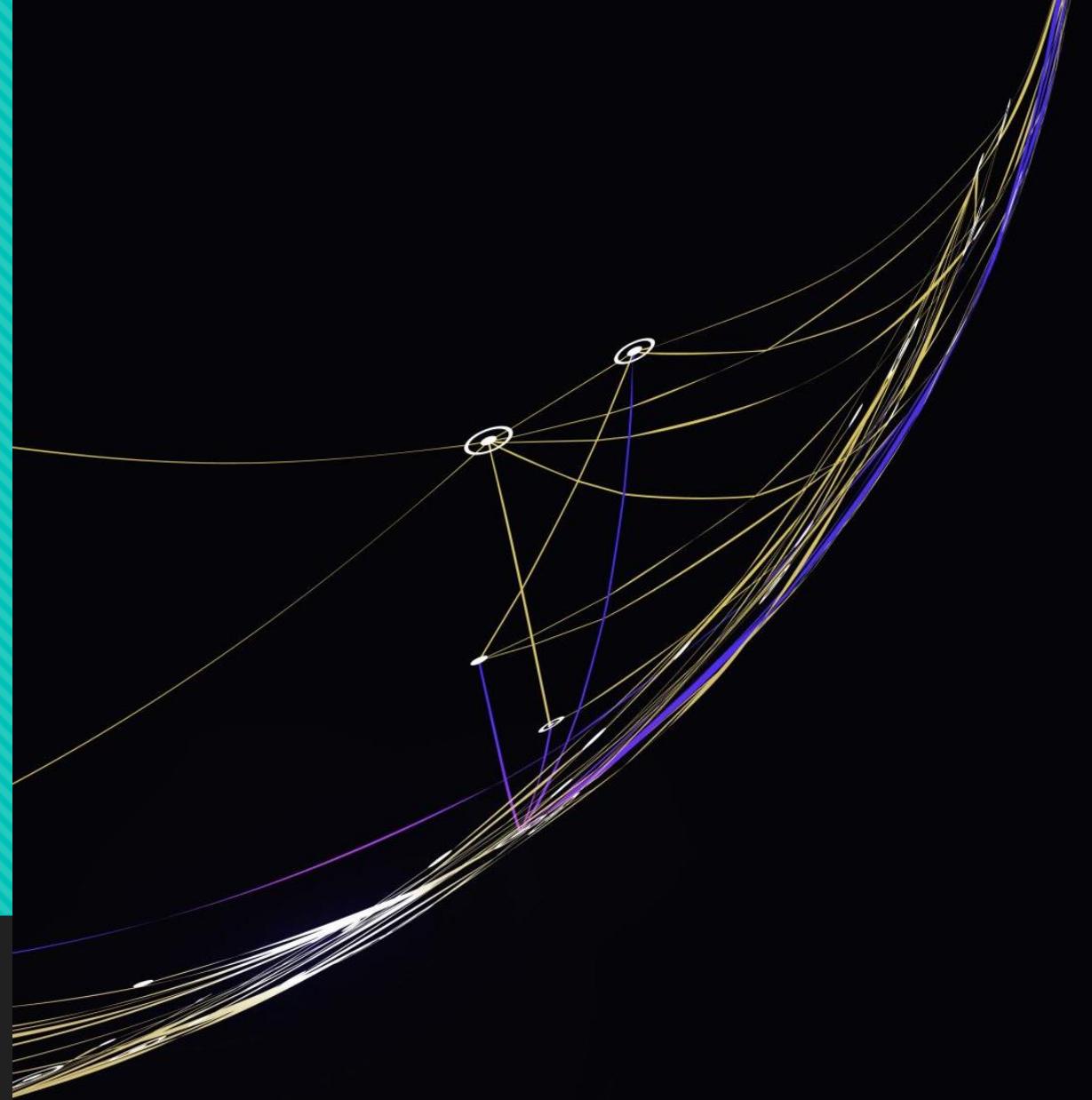


10 minute
break

SDLC Approaches Intro



Why Do We Need An Approach?



Is Agile The Answer?





Benefits of Agile

- Can be rolled back when needed
- Adapts well to a market prone to dynamic change
- You don't need to know what the end-state looks like

Disadvantages to Agile

- Requires constant communication
 - Daily standups
- Can be chaotic for inexperienced leadership

Alternatives to Agile

- Waterfall
- Rapid Application Development (RAD)
- Lean
- Extreme Programming (XP)
- Feature Driven (FDD)



Waterfall

Advantages Of Waterfall

- Easier to manage both the team, and budget
- Works well when change is not expected to come to the market
- Clear deadlines, requirements and tasks



Disadvantages of Waterfall

- Very rigid
 - Demands that each stage of the SDLC be performed regardless of externalities
- Slow to respond to immediate needs



Lean



Advantages Of Lean



Better for adapting to low-budgets



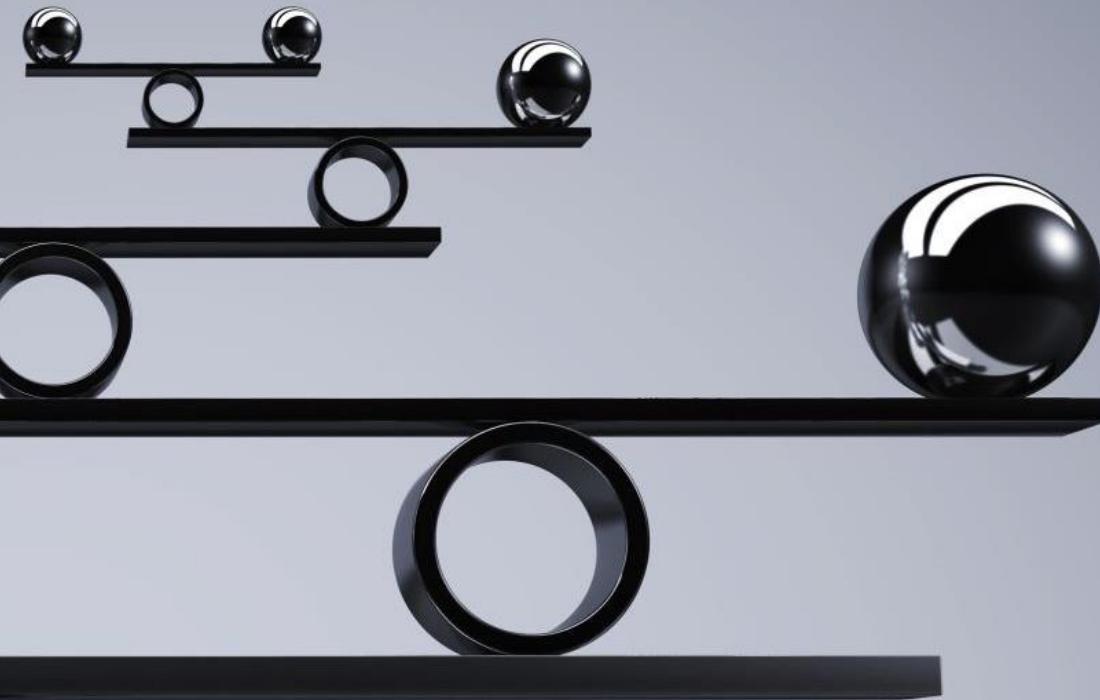
Allows for prioritization of unpredictable needs/rapid market changes



Optimizes tasking to avoid waste of time and resources

Disadvantages of Lean

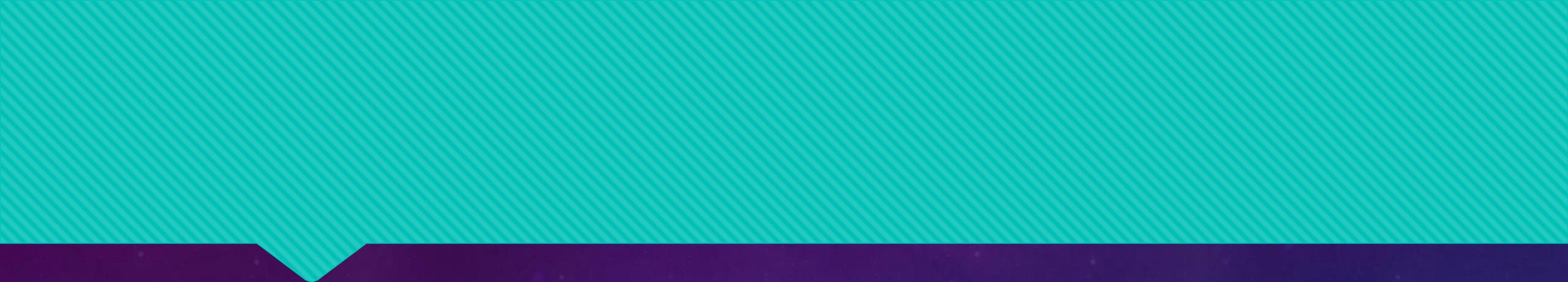
- Very basic approach to SDLC
- Stripped-down approach may not allow enough time to focus on quality feature development



Session Review

Question or
clarifications?





See you next time!