CSci 3081W: Program Design and Development

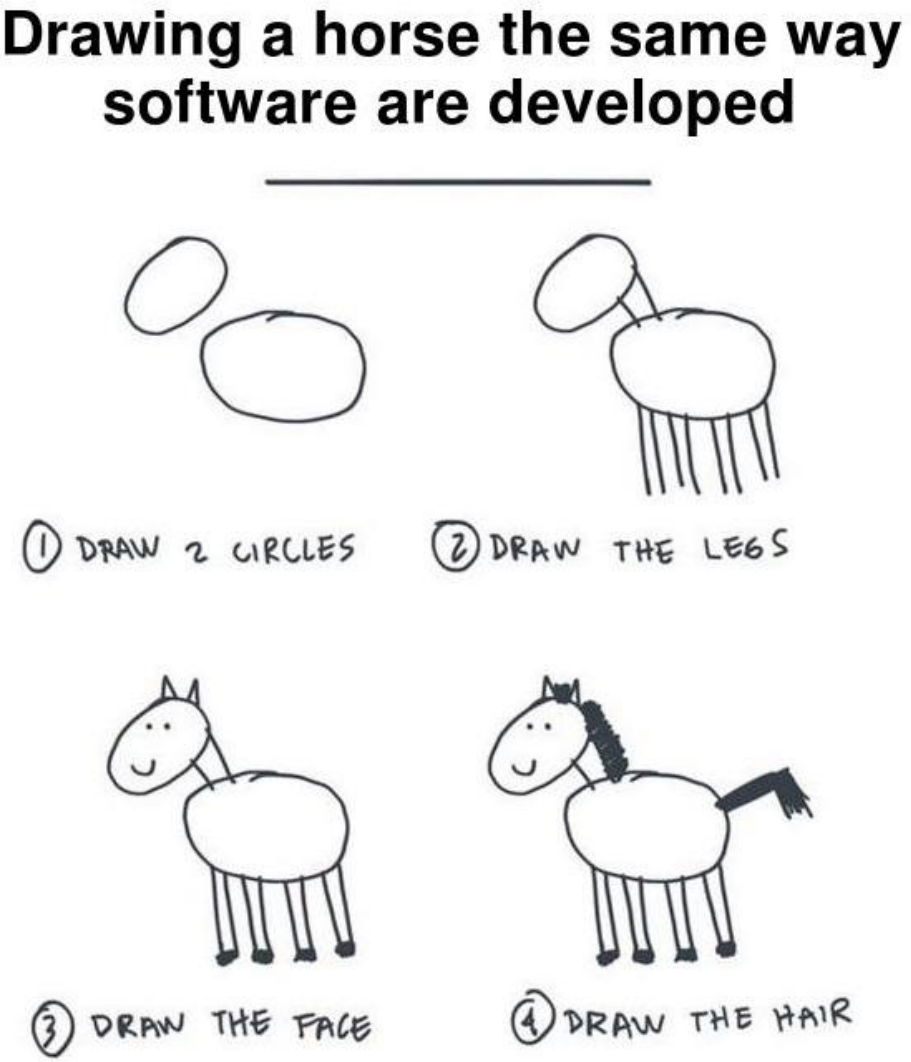
Lecture 14 – Development Processes

Agile

Challenges with waterfall

Agile

Challenges with waterfall

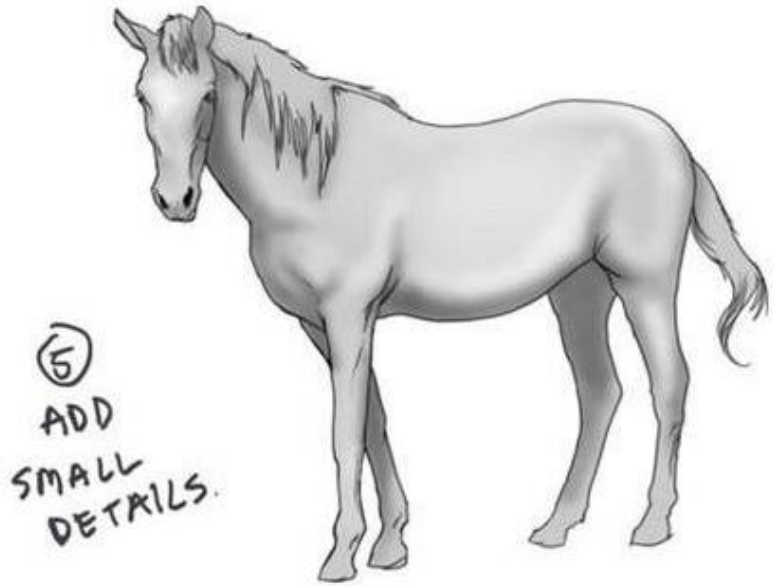


Agile

Challenges with waterfall

client says he had not thought about the requirements properly

boss says developers will only need to add few small details



Agile

Challenges with waterfall

- During verification – pieces don’t work together which leads to an expensive fix
- Users/clients argue that the system doesn’t meet the initial requirements (Predicting user needs is very difficult)
- Potential market shift which also leads to an expensive fix

Agile

17 people, Feb 2001

- Agile Manifesto

Agile is a mindset

- 4 values and 12 principles

Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

Agile Principles

- Highest priority is to satisfy the customer through early and continuous delivery of valuable software

- Welcome changing requirements, even late in the development. Agile processes harness change for the customer’s competitive advantage.

- Deliver working software frequently, preferably a shorter time scale.

Agile Principles

- Businesspeople 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 best way of conveying information to and within a development team is face-to-face

Agile Principles

- 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.

Agile Principles

- Simplicity is essential. (the art of maximizing the amount of work not done)

- 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.

How does Agile solve the challenges of waterfall?

- Adaptive
  - Deliver working software frequently
  - Welcome change
  - Technical excellence and good design
  - Continuous improvement

- Detect translation issues early
- Validate user needs earlier
- Detect Integration issues early

- People and Interaction
  - Business and Developer work together
  - Face-to-Face conversations
  - Self organizing teams
  - Promote sustainable development
  - Motivated individuals

- Detect translation issues early

What new challenges does Agile bring?

- Adaptive
  - Deliver working software frequently
  - Welcome change
  - Technical excellence and good design
  - Continuous improvement
- People and Interaction
  - Business and Developer work together
  - Face-to-Face conversations
  - Self organizing teams
  - Promote sustainable development
  - Motivated individuals

- Architecture/Design/Database modeling is challenging
- Lack of control / Unpredictable Journey - Very uncomfortable for Leaders/Organizations

- Requires participation from customers through out the development process

Scrum

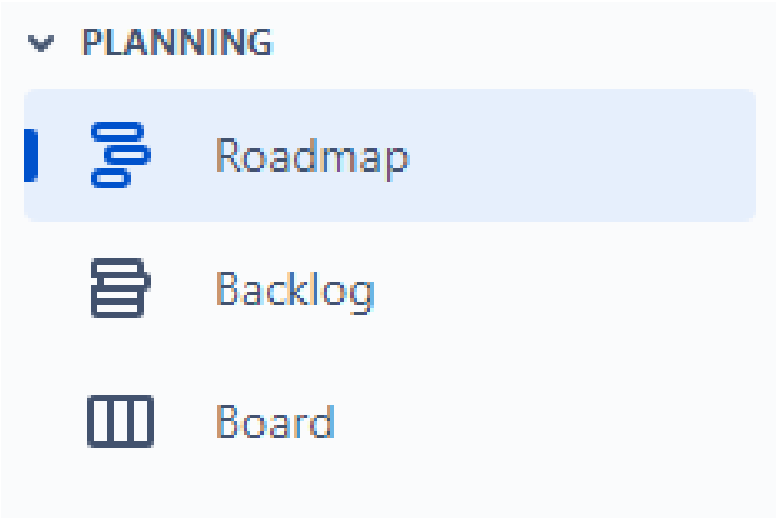
- Incremental and iterative framework for developing, delivering, and sustaining products
- Assume that customers will change what is wanted
- Team
  - Product owner
  - Developers
  - Scrum master

Scrum

https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf

Workflow

- Sprint
- Sprint planning
- Daily scrum
- Sprint review
- Sprint retrospective
- Backlog refinement



Scrum

Some alternative implementations are

Kanban

Scrumban

Kanplan

Scrum

Workshop 6 – Atlassian’s Jira

More Software Development Models

Spiral

Rational Unified Process

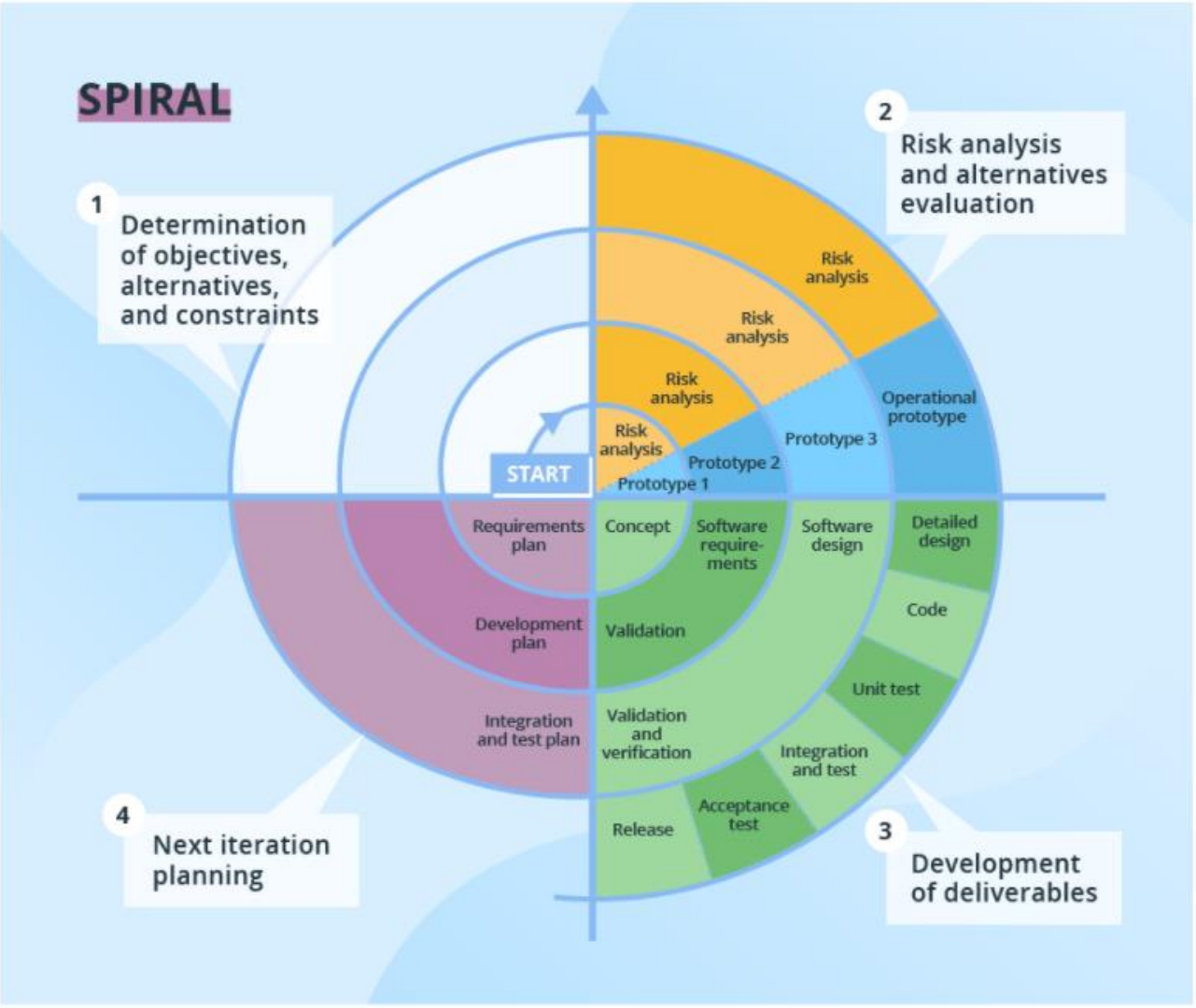
Extreme Programming

Spiral

Iterative development with systematic controlled aspects of waterfall

High emphasis on risk analysis

Potentially allows incremental releases (or updates) each spiral

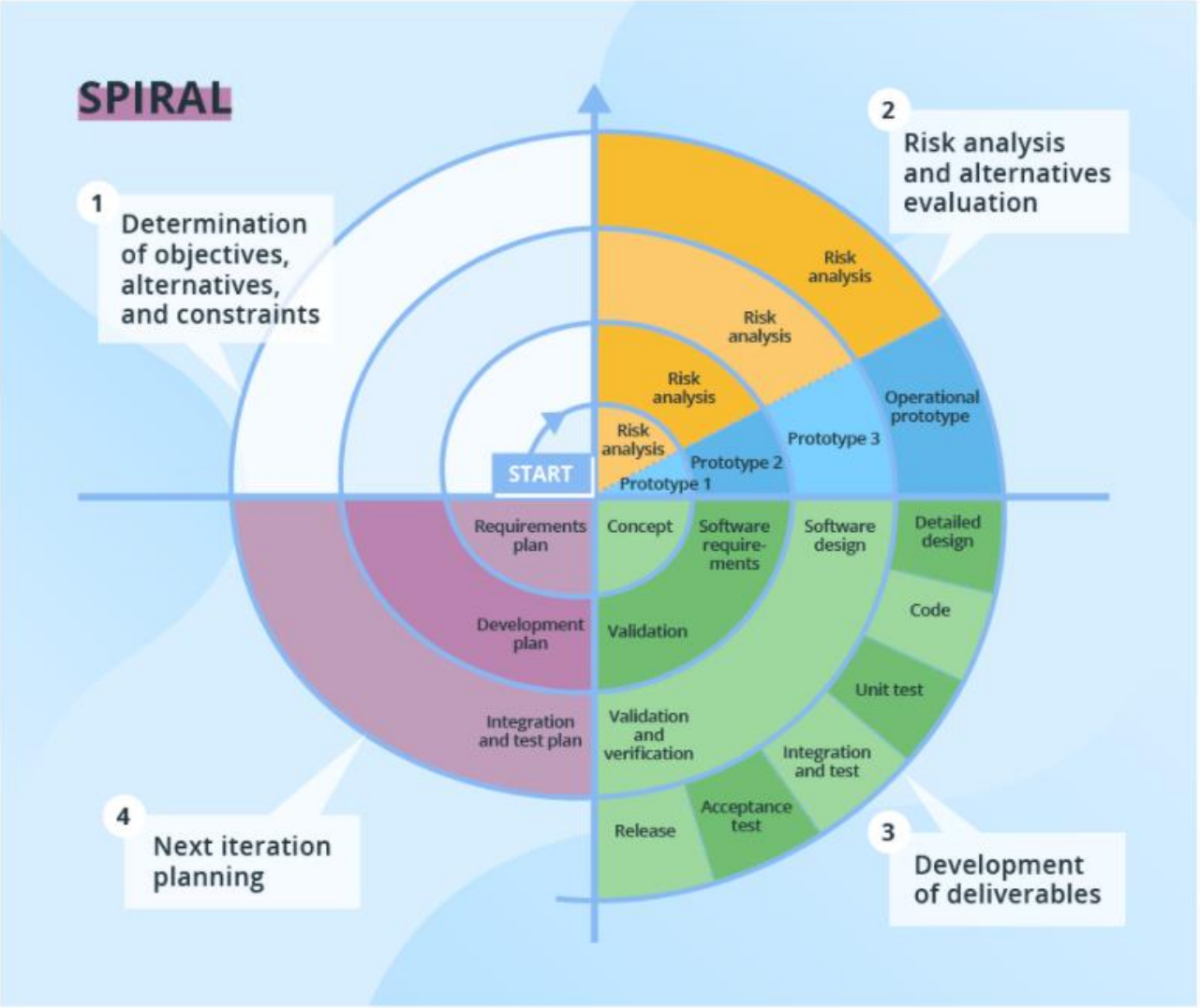


Spiral

Changing requirements can be adapted in

Requirements can be captured more accurately

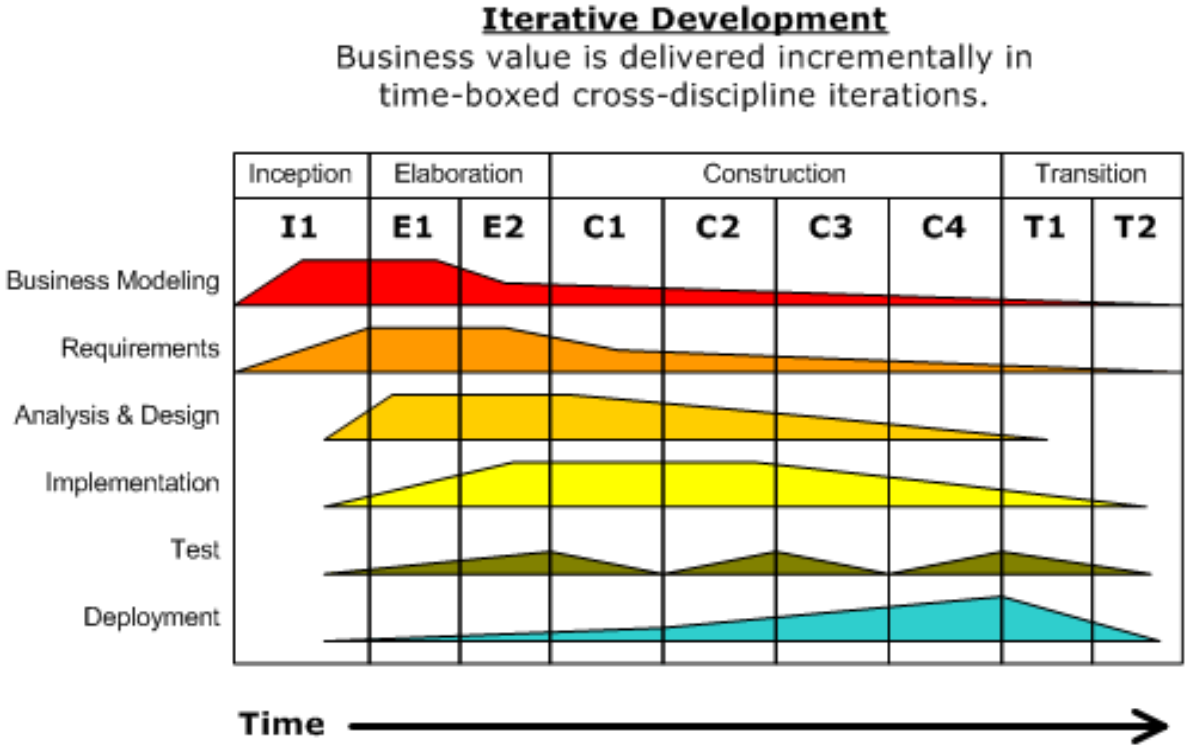
Huge emphasis on risk management



Rational Unified Process – by IBM

Four life cycle phases

- Inception
- Elaboration
- Construction
- Transition

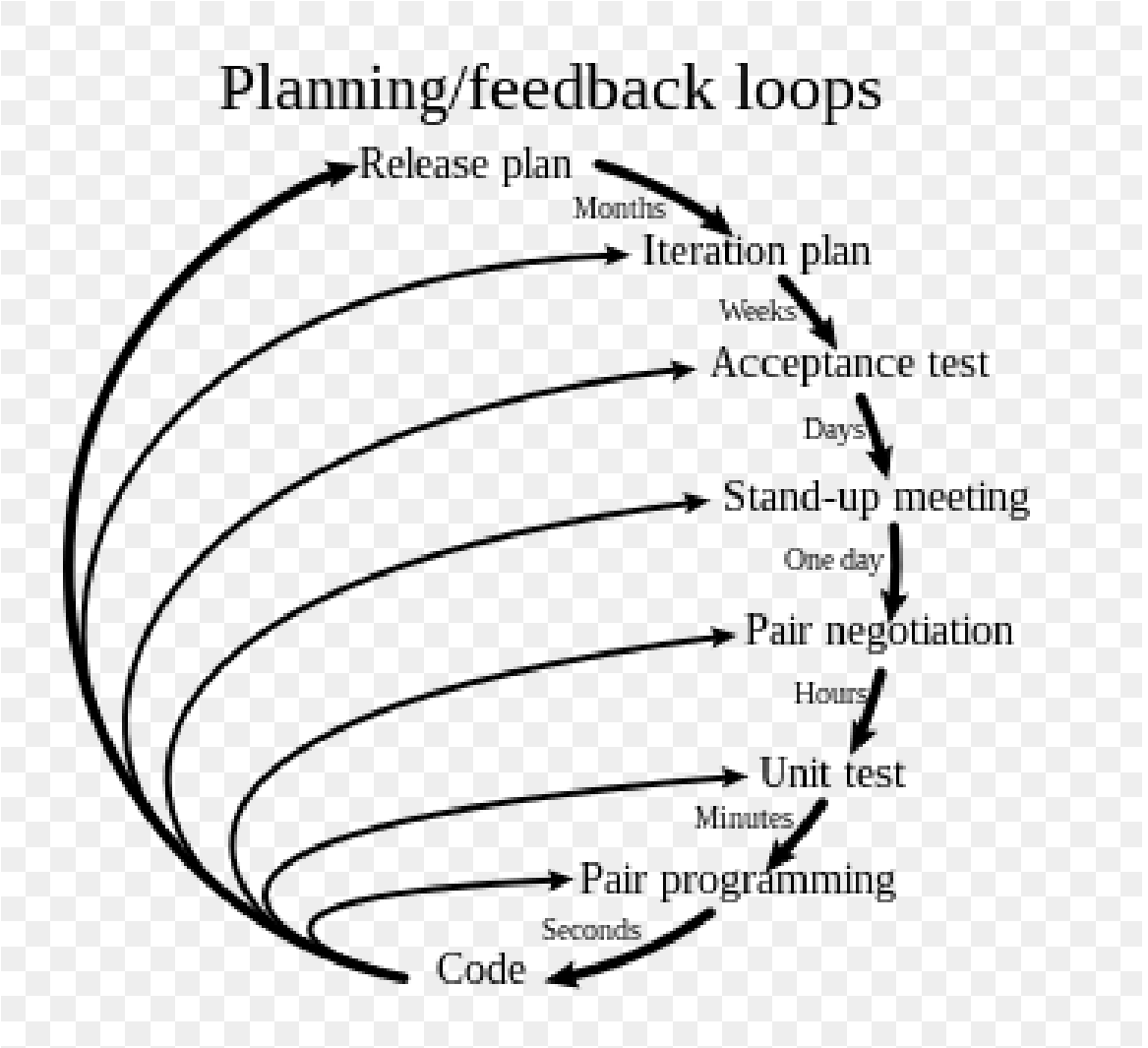


Extreme Programming

Type of agile software development

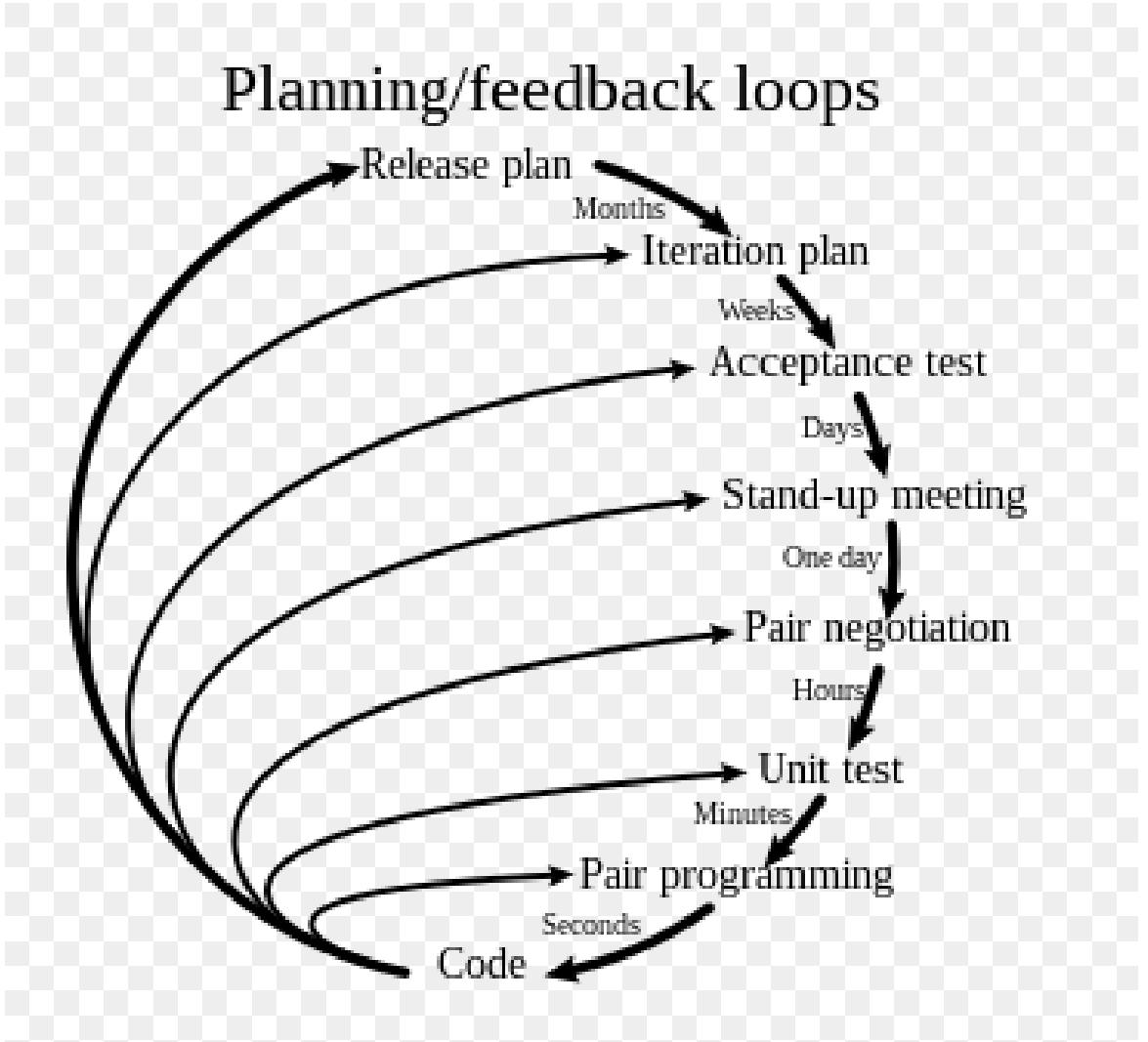
Working in pairs

Unit testing of all code and acceptance testing of all requirements



Extreme Programming

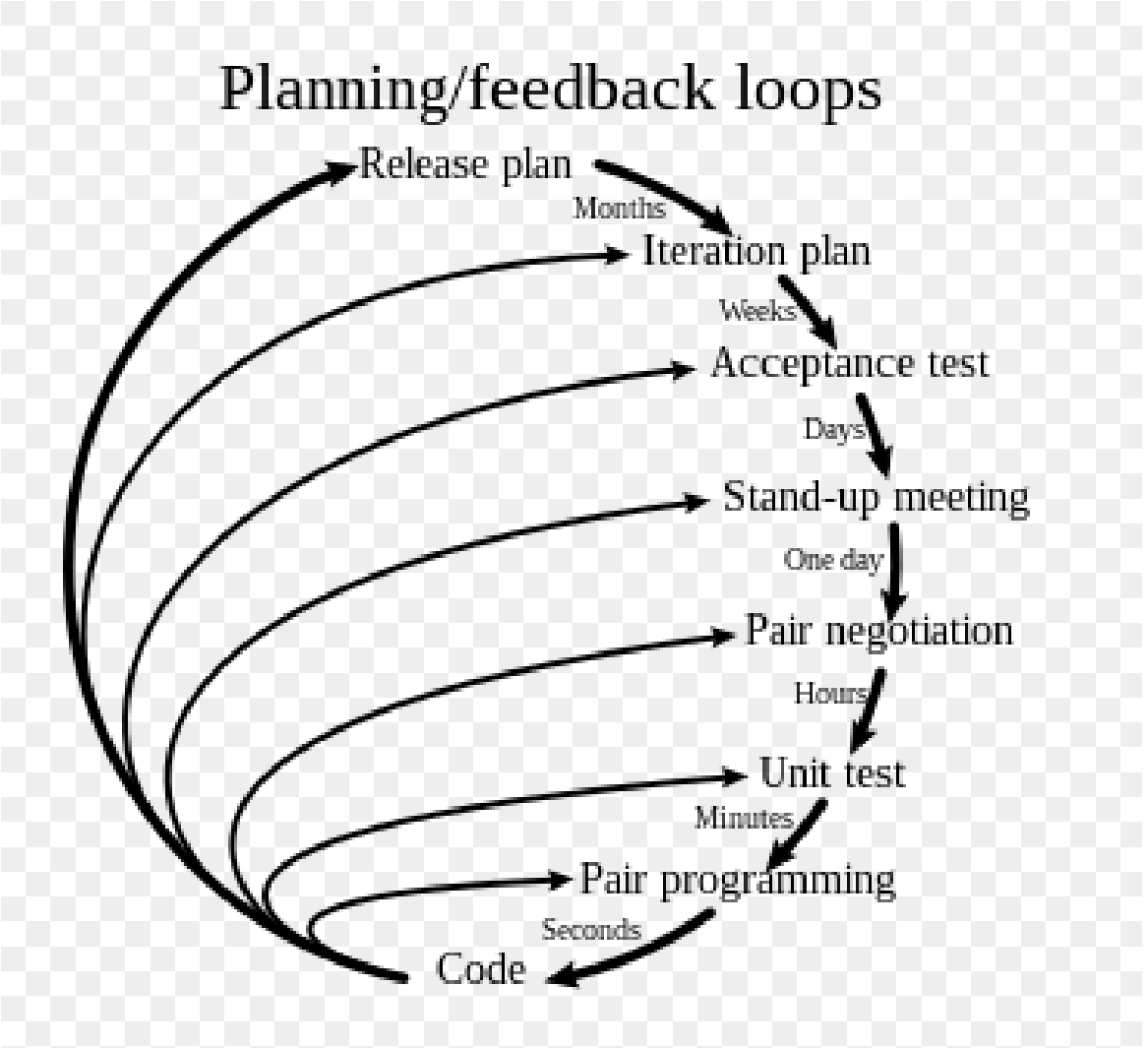
Don’t program something until it’s needed



Extreme Programming

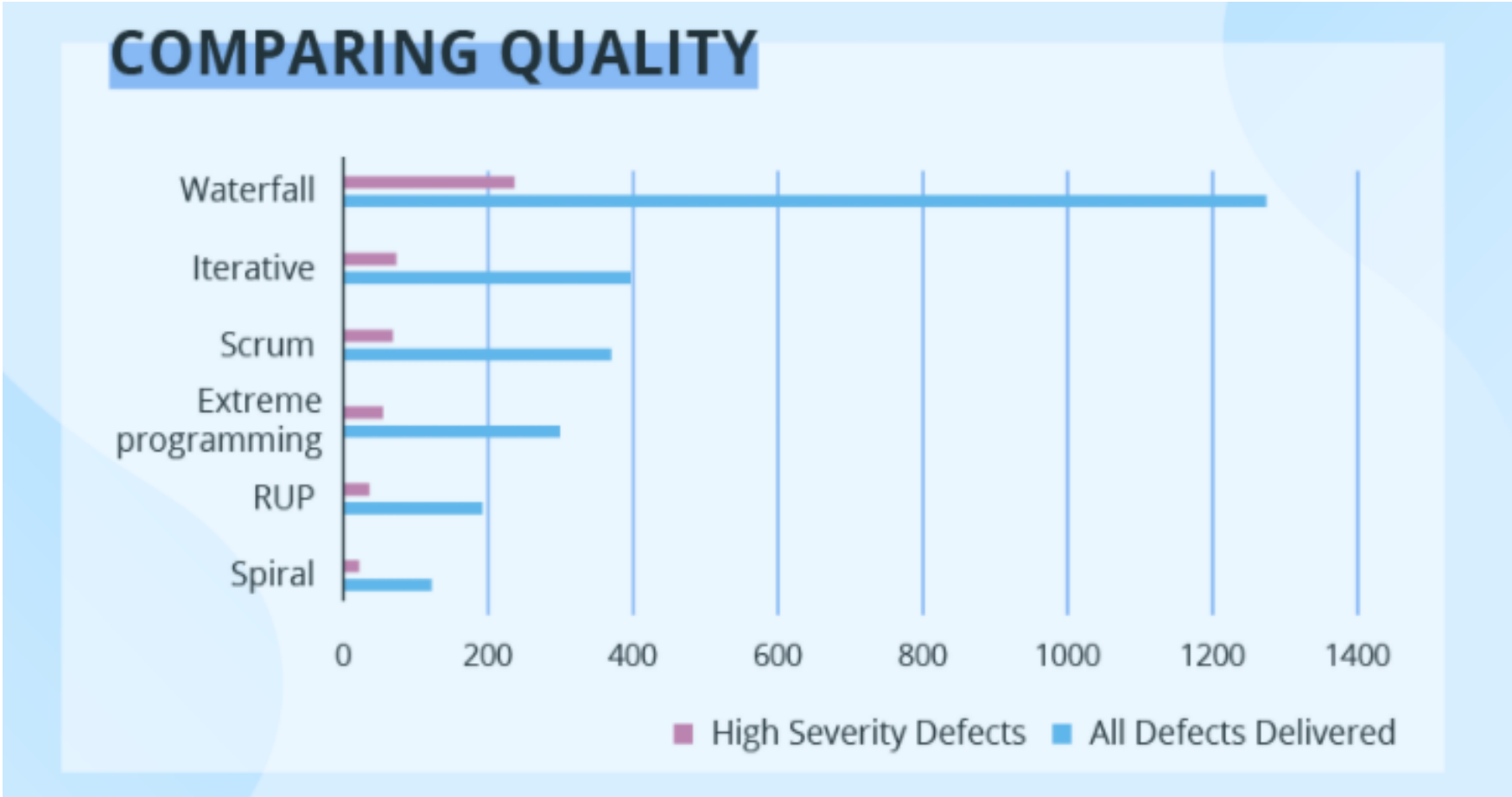
Four activities:
Coding
Testing
Listening
Designing

29 rules total
Unit tests first (TDD)
Customer is always available



Comparing Costs

Comparing Quality



Comparing Total Cost of Ownership over 5 years


Using Jira doesn't make you agile
If those project managers could read, they'd be very upset.