



Software Development Process

Agile Development Methods

Team Teaching Software Engineering Courses
Department of Information Technology
Malang State Polytechnic

Objective

- To explain Agile Development Methods which is a software development methodology
- Agile Development Methods software development processes and models

Agile Development Methods

The Agile Model is a software development process that developed in 1990. This methodology, known as agile development methods, prioritizes flexibility regarding changes that occur during development. Agile models include Rational Unified Process (1994), Scrum (1995), Crystal, Extreme Programming (1996), and Adaptive Software Development, Feature Driven Development, and Dynamic Systems Development Method (DSDM) (1995). And finally, in 2001, the Agile Modeling (AM) development process was formed.

Agile Modeling (AM) Development Process

Done in iteration or repetition. If a software development project is carried out using the agile method, then during the work period there will always be an iterative development process.

12 Principles of Agile Development Methods

- The top priority is to satisfy clients by producing valuable software quickly and regularly
- Ready for changing needs. Agile processes leverage change for the client's benefit
- Produce software that works routinely, from a few weeks to several months, with an emphasis on short time periods
- Business partners and software developers must work together throughout the project
- The project development environment has a motivational atmosphere. Give them the environment and support they need, and trust them to get the job done well
- The most efficient and effective method for exchanging information from and within a development team is by direct communication



- Working software is the ultimate measure of a team's progress
- Agile processes support continuous development with a consistent development pace
- Attention to technical and design details will increase agility
- Simplicity (maximizing the amount of work yet to be done) is very important
- Self-organizing teams support good software architecture, requirements, and design
- Periodically, the development team reflects on how to make development more effective, then adjusts the way they work.

Benefits of Agile development methods

- High-value & working App system
- Iterative, incremental, evolutionary
- Cost control & value-driven development
- High-quality production
- Flexible & risk management
- Collaboration
- Self-organizing, self-managing teams

High-value & working App system



Software can be produced that has a high selling value, manufacturing costs can be reduced and the software can run well

Iterative, incremental, evolutionary



The team must work in a short time (usually 1-3 weeks) and also always add functionality to the software according to the client's needs

Cost control & value-driven development

Software development is tailored to user needs, the team can quickly respond to user needs so that the time and costs of creating software can be controlled

High-quality production

The cost of creating software can be reduced and the manufacturing process can be accelerated, but the quality of the software created must be maintained

By testing each software functionality after it has been created, Agile also accommodates this need

Flexible & risk management

If you use the manufacturing method that is usually used and if you want to change the functionality of the wireframe that has been created, it requires a complicated process

Starting from meetings with system analysts to change software systems, changes to product release plans to changes in production costs so that software functionality is easily changed and ultimately software failures can be minimized.

Collaboration

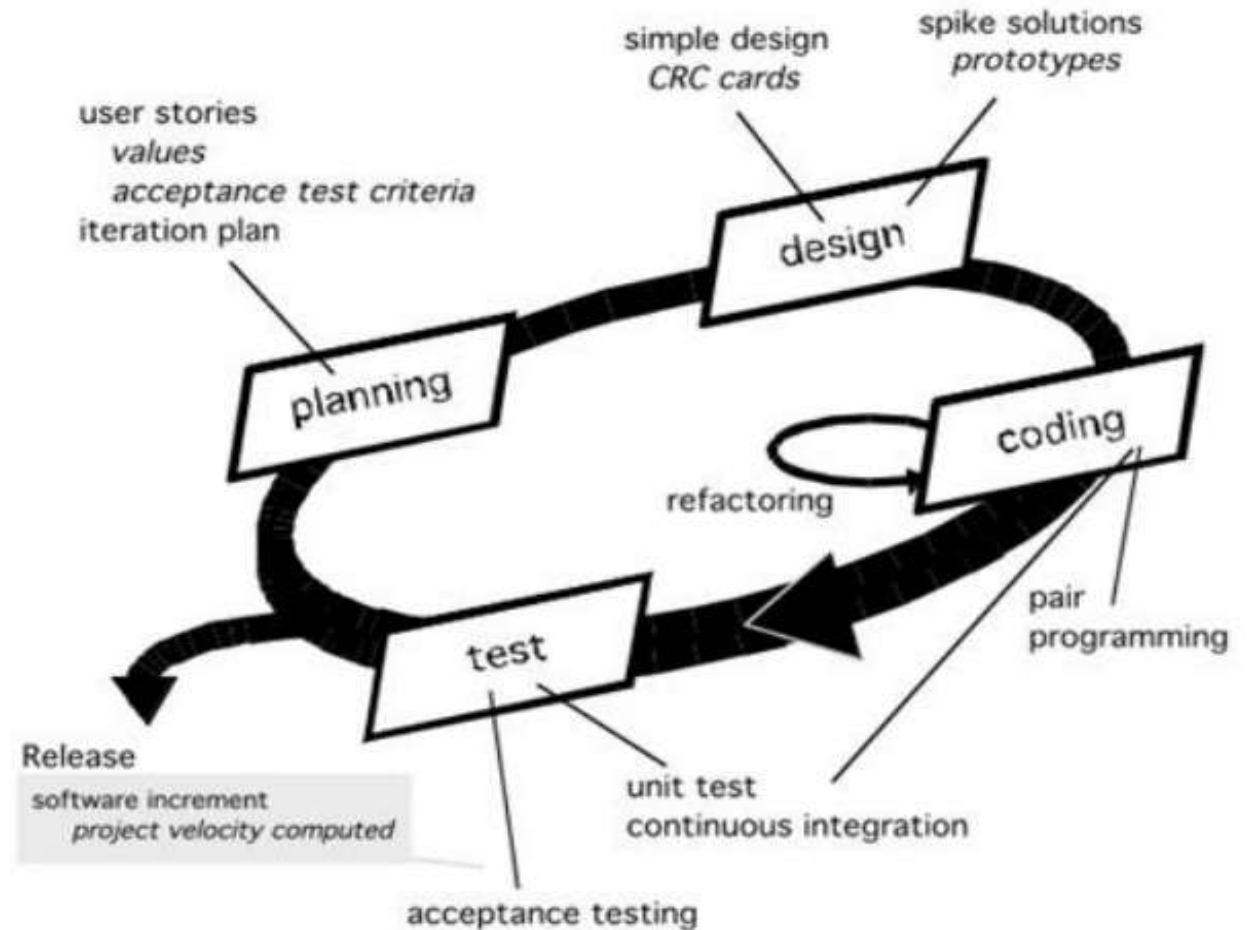
The development team is required to meet frequently to discuss project progress and feedback from clients which will later be added to the software, so that the team can collaborate optimally

Self-organizing, self-managing teams

Hire the best people, provide and support their needs and then let them work. That is the difference between agile and other HR. With agile, developers can manage themselves, while team managers are only tasked with collaborating software developers with clients. So a solid team is created

Agile XP Model (eXtreme Programming)

This model not only carries out the code and testing process, but also carries out the process of mapping the system architecture so that the coding and testing process can be carried out without disturbing the existing system. The ongoing system analysis process is carried out in order to better understand the system workflow

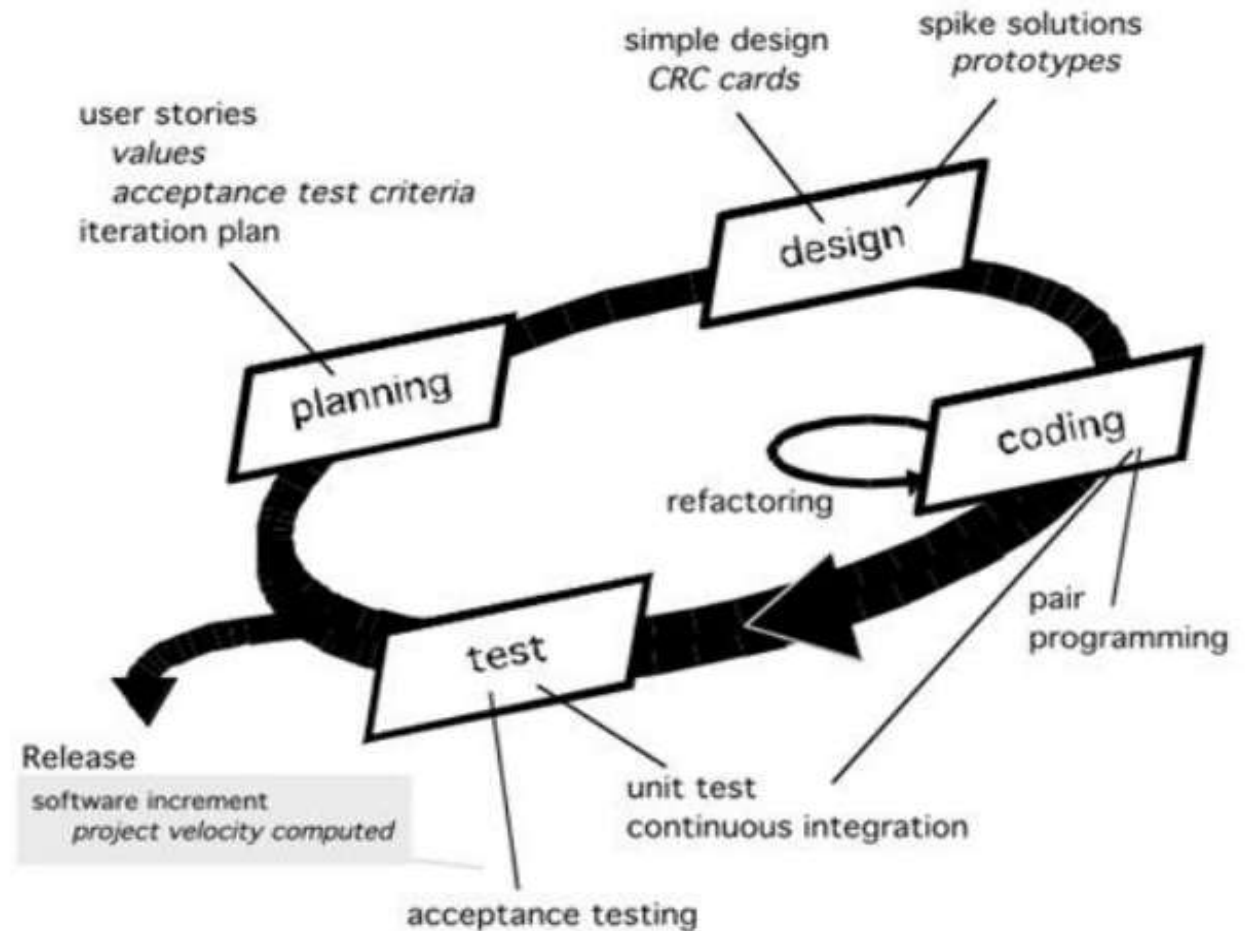


Stages - Agile Model

1. Planning

Planning is a methodical process designed to achieve specific goals and decision making to achieve desired results. Requirements needed at this stage of data collection techniques:

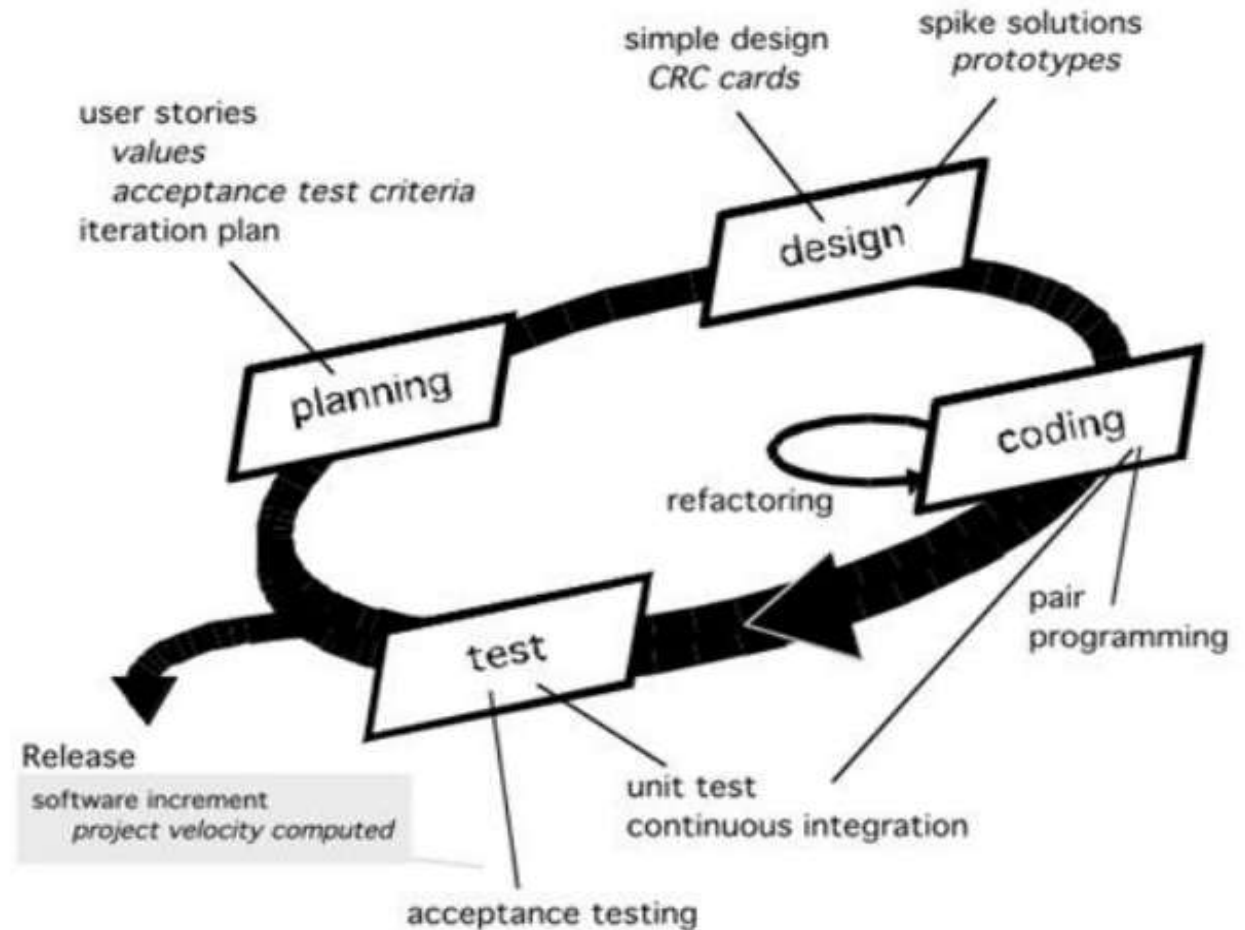
- Analysis of system requirements
- Identify actors
- Identify use cases



Stages - Agile Model

2.Design

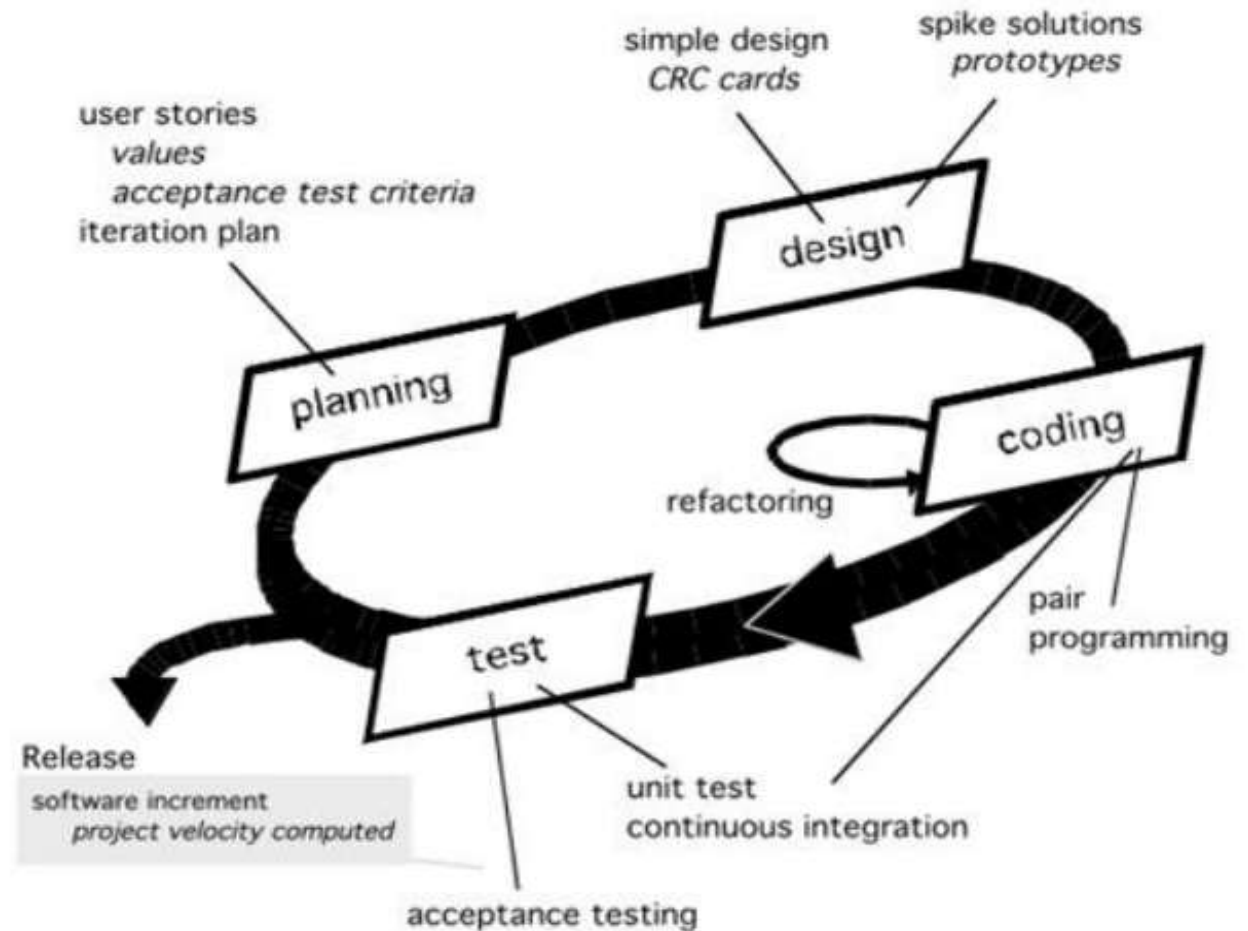
Design activities in application development aim to organize logical patterns in the system. A good design can reduce dependency between each process in a system. That way, if one feature on the system is damaged, it will not affect the system as a whole



Stages - ModelAgile

3.Coding

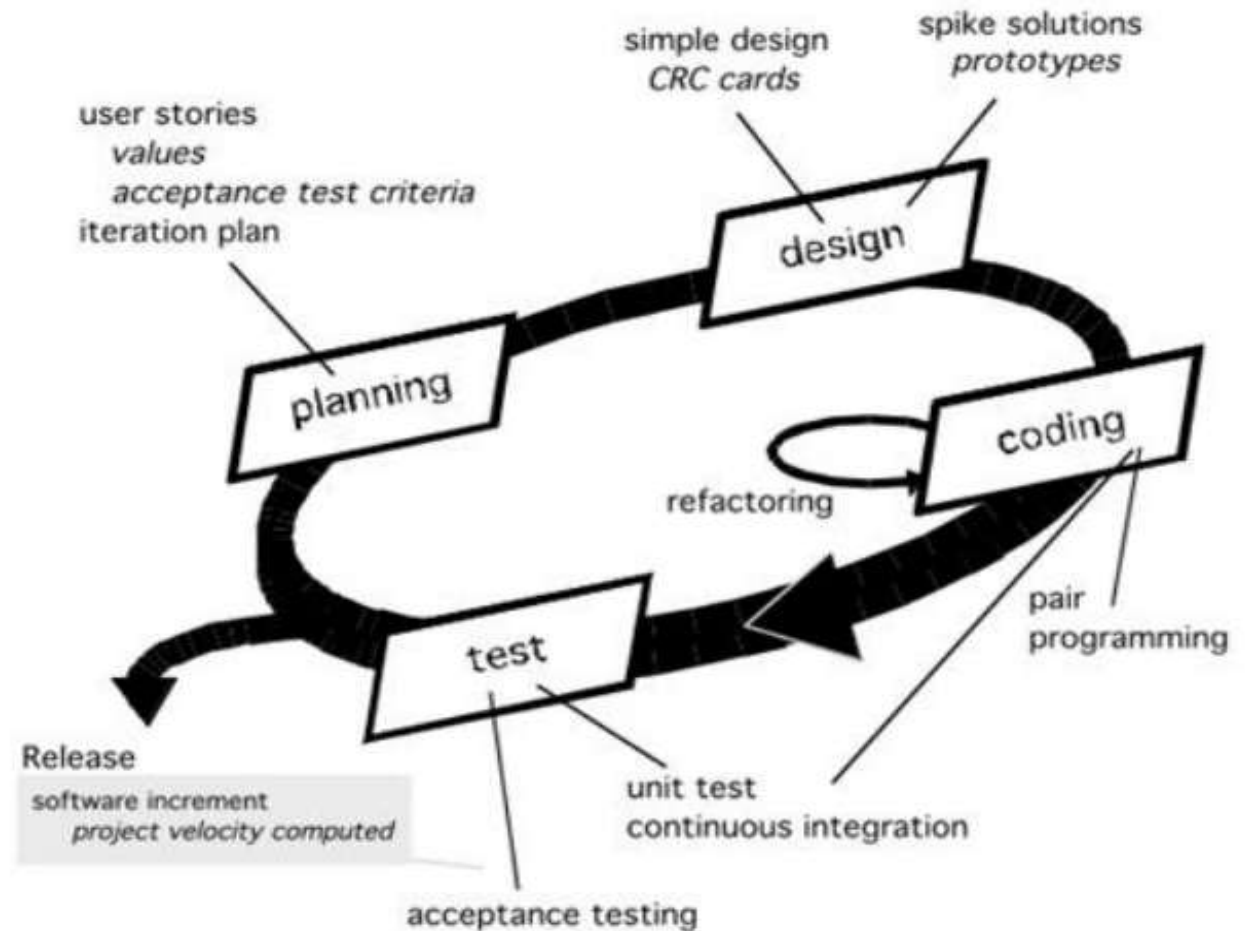
After the various unit tests have been built, the team then continues its activities with writing application coding. XP applies the pair programming concept where each task of a module is developed by 2 programmers



Testing

4. Testing

At this testing stage, the application is directly tested by the user or client and receives direct feedback regarding the application of the story described previously



Advantages of XP

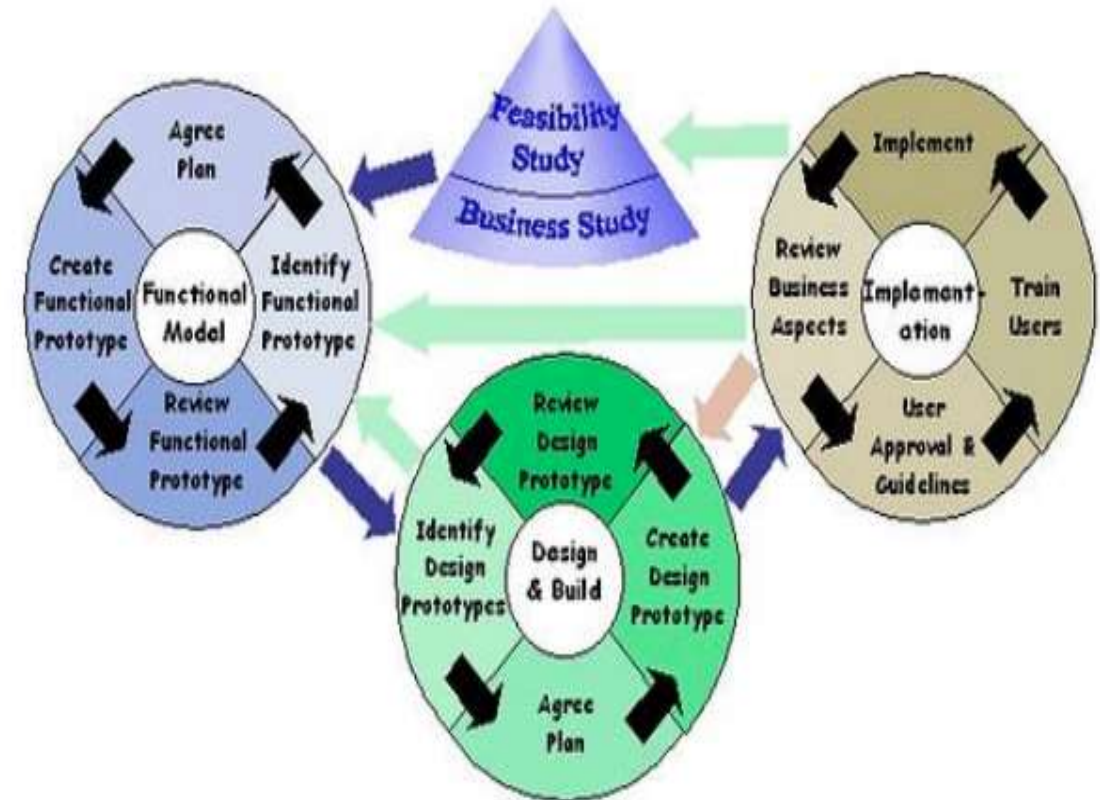
- Immediate feedback after receiving a response from the user
- Collective ownership which does not depend on one person for work
- Changes and additions to requirements can be responded to even though the development process is almost complete
- A development process that includes one person from the user side as an on-site customer makes communication easier during the development process
- A popular method because it is more relaxed and non-restrictive
- Cheaper costs
- Able to automate tests
- Respond to every feedback by carrying out tests, unit tests or system integration and don't delay because costs will increase (money, energy, time)
- Lots of new ideas and dare to try them, dare to rework and every time an error is found, it is immediately corrected

Lack

- Developers must always be ready for change because change will always be accepted
- Cannot create detailed code at the start (simplicity principle and also the recommendation to do what is needed that day)
- XP has no formal documentation created during development. The only documentation is the initial documentation carried out by the user
- Communication is always oral without formal documentation

DSDM Agile Model (Dynamic Systems Development Method)

Dynamic System Development Method provide framework for building and maintaining systems in a limited time through the use of incremental prototyping in a conditioned environment

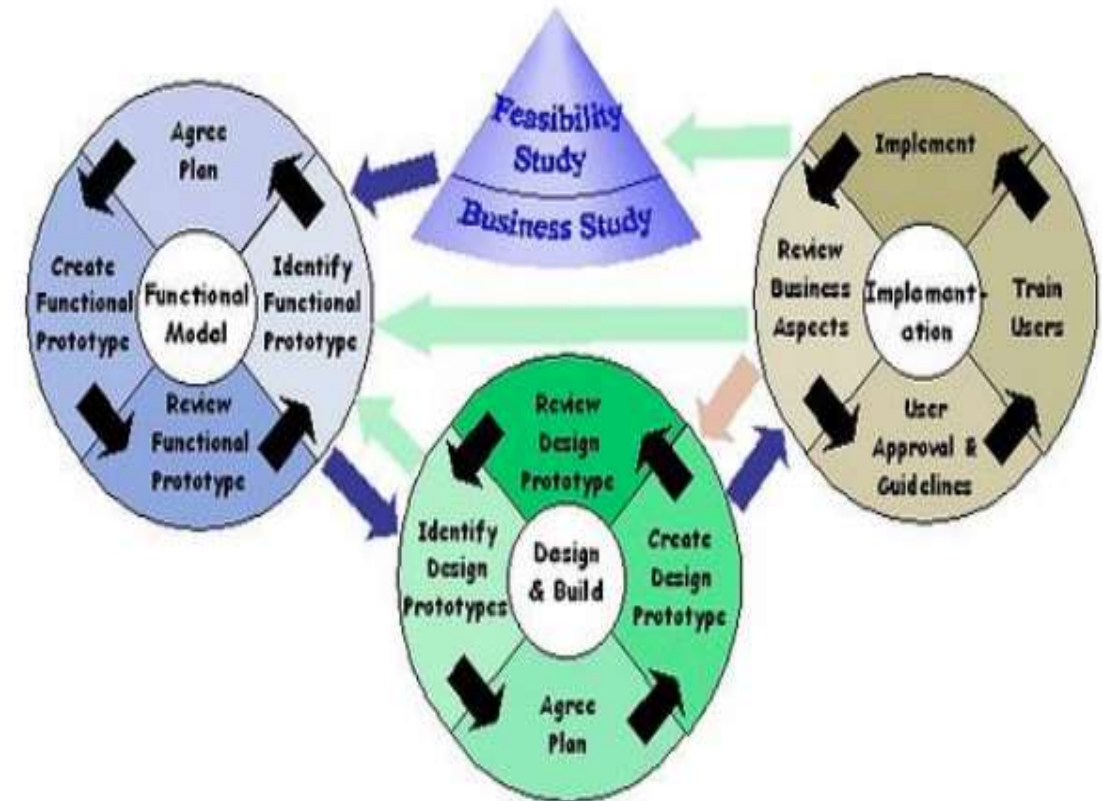


DSDM Agile Model (Dynamic Systems Development Method)



1. Feasibility Study

Is a high-level feasibility report that allows the project steering committee to decide on the future of the project, and further feasibility studies

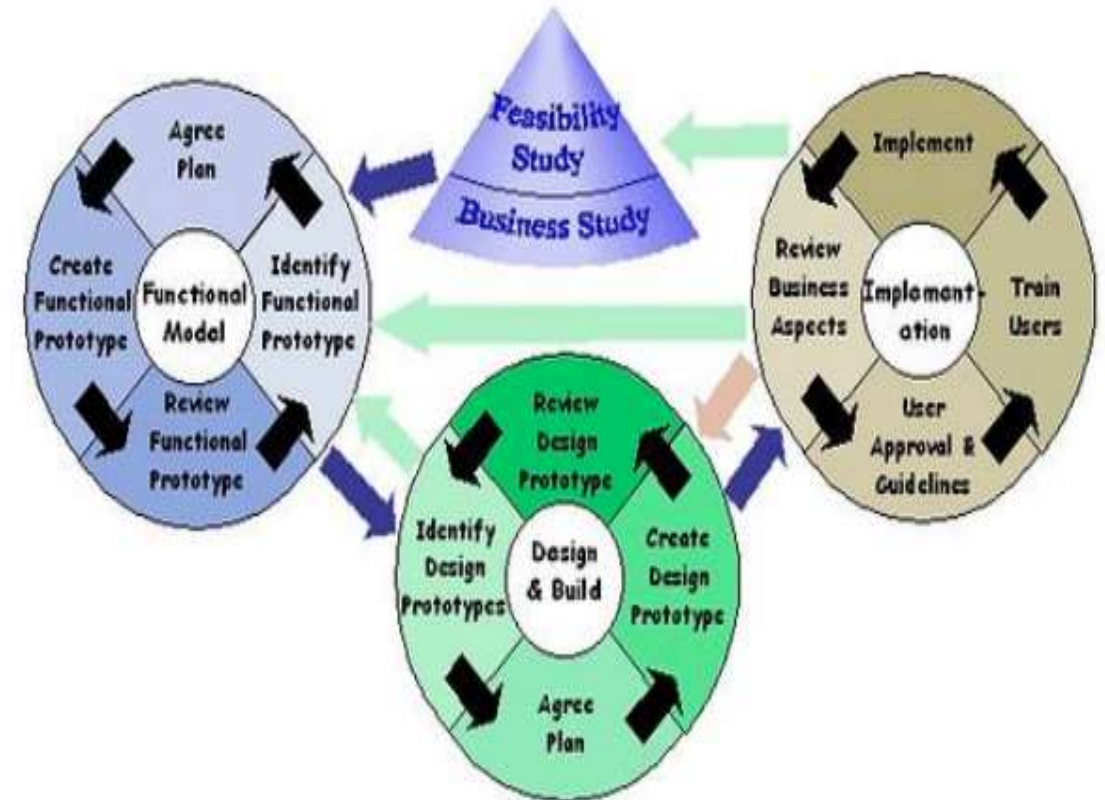


DSDM Agile Model (Dynamic Systems Development Method)



2. Functional Model

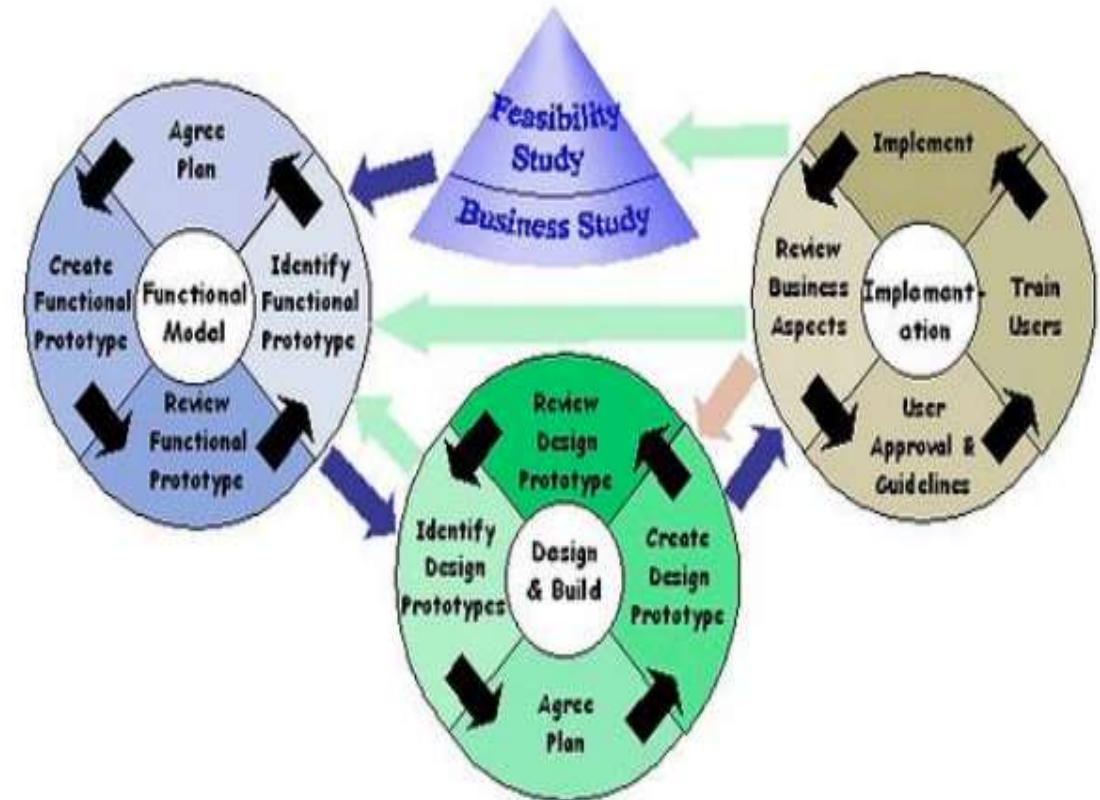
Functional models consist of both working software prototypes and static models. This phase results in the processing of information obtained in business research



DSDM Agile Model (Dynamic Systems Development Method)

3. Design and Build Iteration

This phase perfects the functional prototype developed to meet functional requirements (Modules), namely: Identify, Plan, develop, validate functions

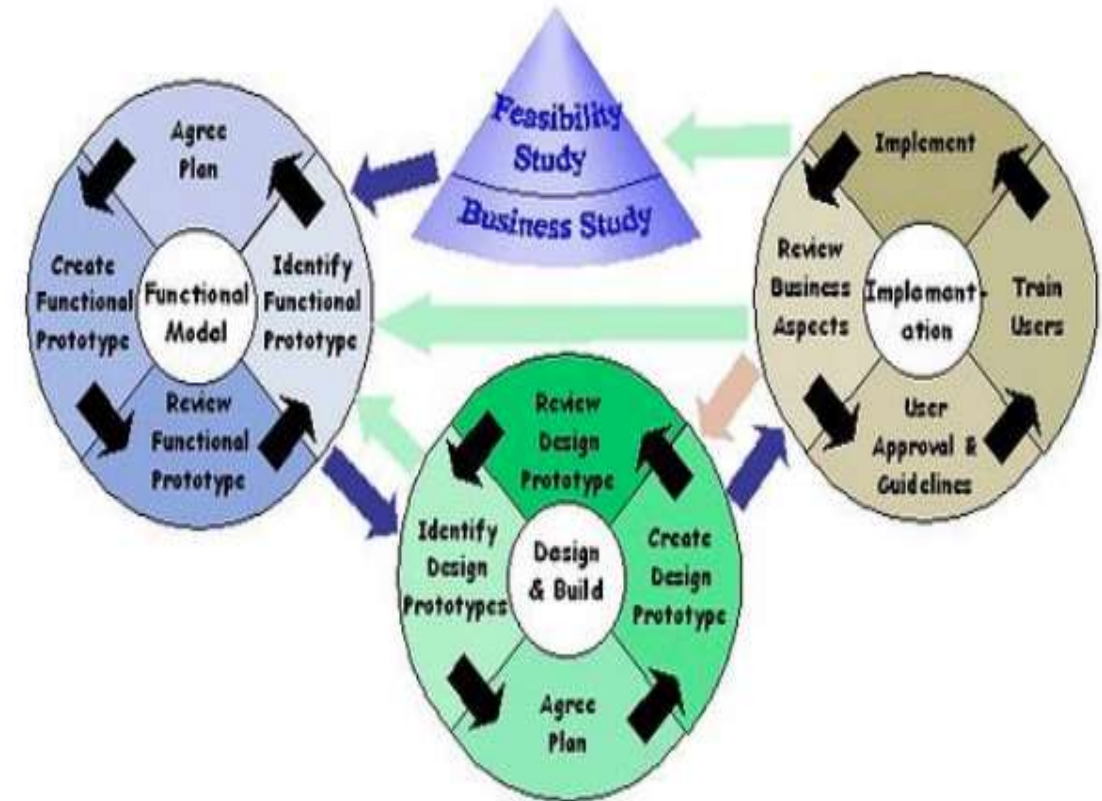


DSDM Agile Model (Dynamic Systems Development Method)



4. Implementation Phase

After various unit tests, once built, the team then continues its activities with writing application coding. XP applies the pair programming concept where each task of a module is developed by 2 programmers.



Excess**DSDM** (Dynamic Systems Development Method)



- This method will build software quickly, 80% of the project is delivered in 20% of the total time to deliver the complete project
- Dynamic System Development Method can be combined with eXtremeProgramming resulting in a combination of process models that follow the Dynamic System Development Method and practices that are in line with eXtremeProgramming

LackDSDM (Dynamic Systems Development Method)



- Business Studies provides the basis for all subsequent product development
- This phase looks at the affected business processes in detail and the information they require



Conclusion

Agile means being fast, light, free to move. So when creating software using agile development methods innovation and innovation are needed responsibility between the development team and the client so that the quality of the software produced is good and the agility of the team is balanced

The Agile Manifesto is the values used to underlie the ongoing Agile Software Development

Agile Software Development, interacting with personnel is more important than processes and tools, working software is more important than complete documentation, collaboration with clients is more important than contract negotiations, and responsiveness to change is more important than following a plan

Agile Development Methods were developed because in traditional methodologies there are many things that make the development process unable to be successful according to user demands.

This methodology has been developed quite a lot, one of which is Extreme Programming (XP), Adaptive Software Development (ASD), Dynamic Systems Development Method (DSDM), Scrum Methodology, Crystal, Feature Driven Development (FDD), Agile Modeling (AM)



SCRUM

Objective

- Students are able to explain the concept of the Scrum framework which consists of
 - Roles
 - Events
 - Artifacts

Scrum in 100 words

- Scrum is an Agile process that allows us to focus on *delivery* with the highest business value in the shortest time.
- Scrum allows us to inspect software that is created quickly and repeatedly (every 2 weeks – once a month)
- Personnel from the business side determine priorities. The team independently determines the best path to *deliver* the features with the highest priority.
- Every 2 weeks – 1 month, everyone can see the results of their work in the form of usable software, which can then be decided to be released or continued development in the next sprint.

Scrum Has Been Used By:

- Microsoft
- Yahoo
- Google
- Electronic Arts
- IBM
- Lockheed Martin
- Phillips
- Siemens
- Nokia
- Capital One
- BBC
- Intuit
- Nielsen Media
- First American Real Estate
- BMC Software
- IPswitch
- John Deere
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting
- Océ

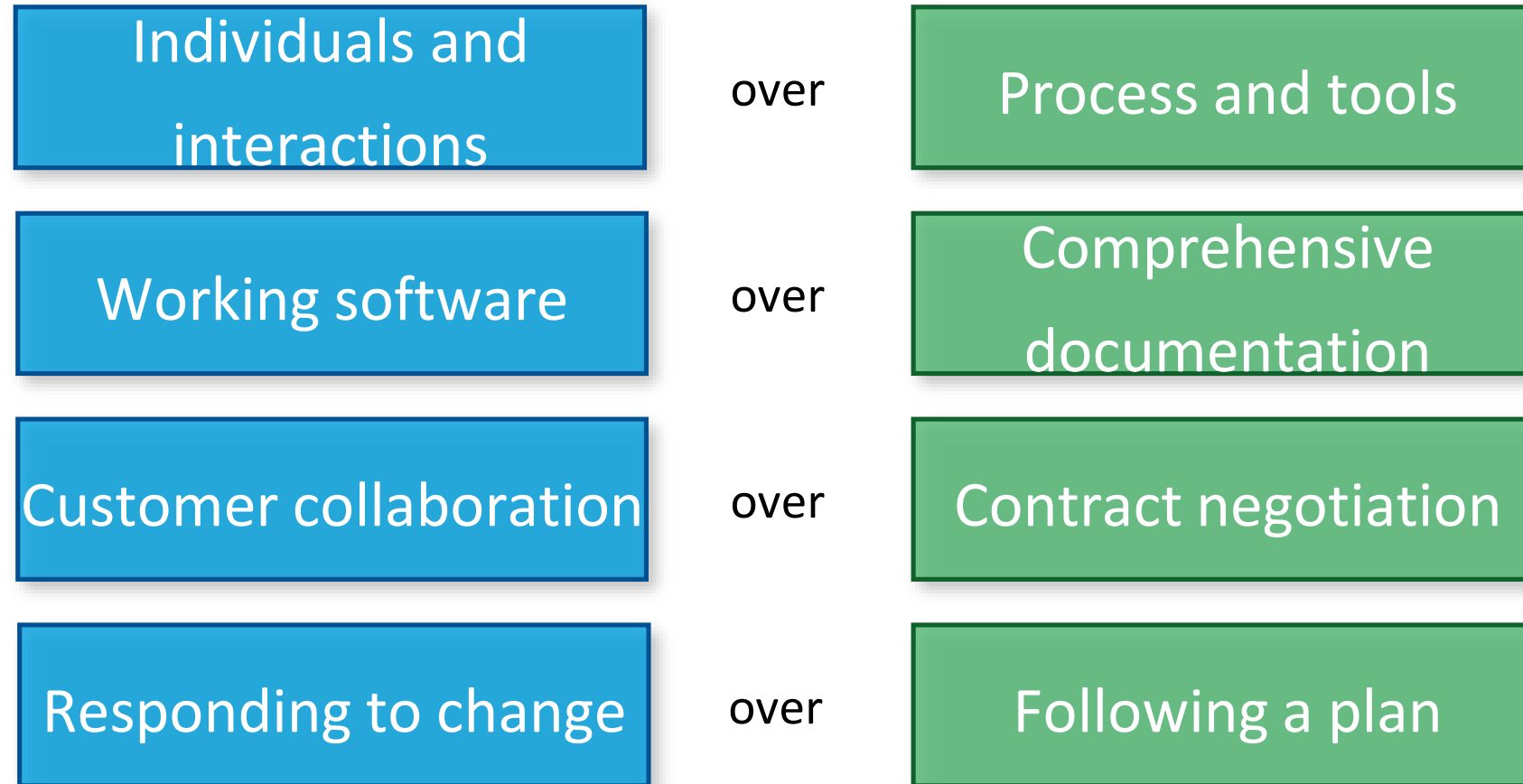
Scrum Has Been Used For:

- Commercial software
- In-house development
- Contract project
- Project *fixed-price*
- Financial applications
- ISO 9001 certified application
- Immersed system
- 24x7 system with 99.999% *uptime*
- Project *Joint Strike Fighter*
- Making *video games*
- FDA-approved, life-critical systems
- Satellite control software
- Website
- Handheld software
- Mobile phones
- Network switching applications
- ISV applications
- Some of the big applications that we know

Characteristics

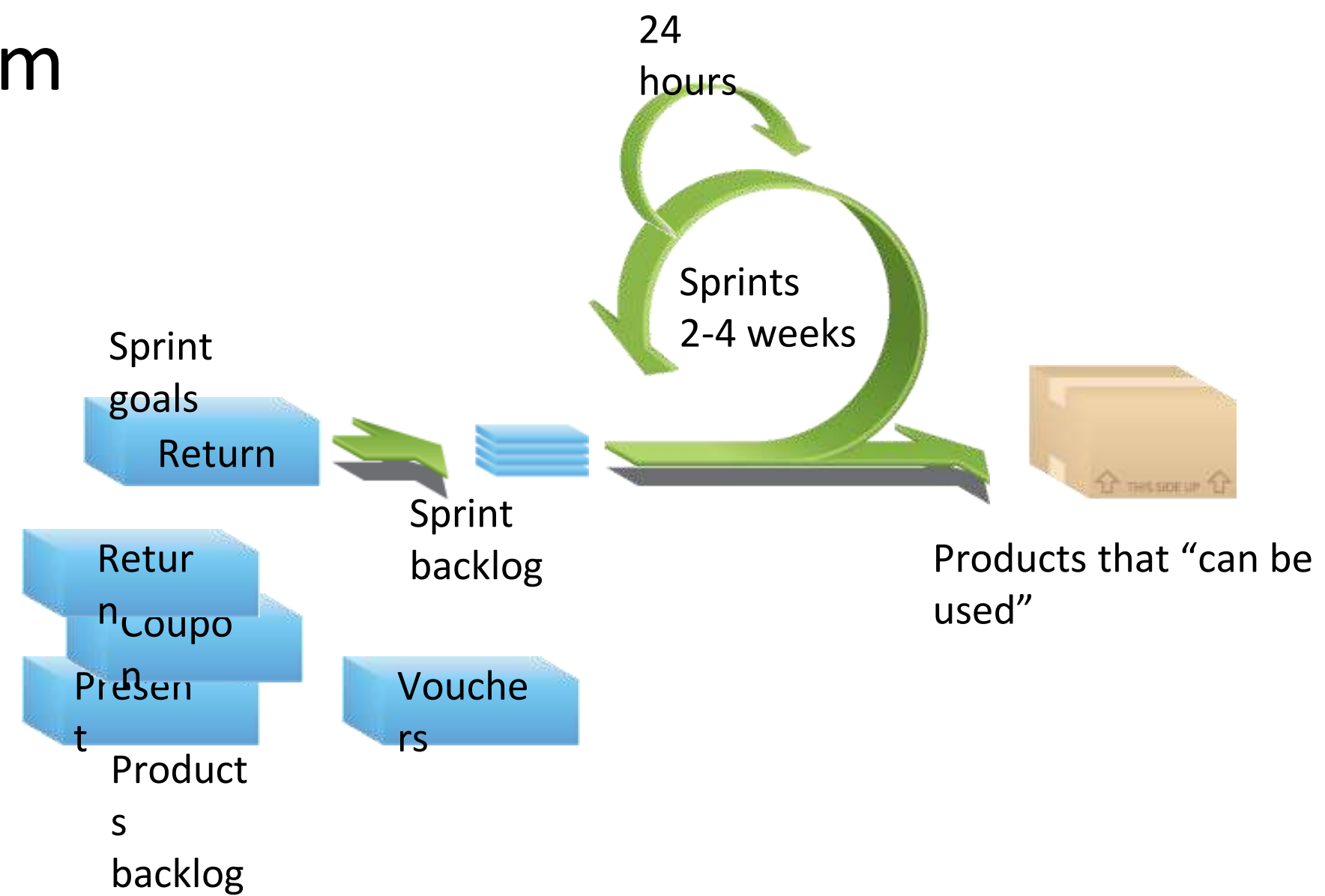
- independent team, *self-organized*
- Products are processed in a series of activities lasting 1-4 weeks called "Sprints".
- System requirements are compiled as items in a list referred to as "*Product Backlog*".
- Deliberately made not too rigid, detailed and technical.
- Using certain generative rules to create an "agile environment" in project completion.
- Is one of the "agile processes"

Agile Manifesto

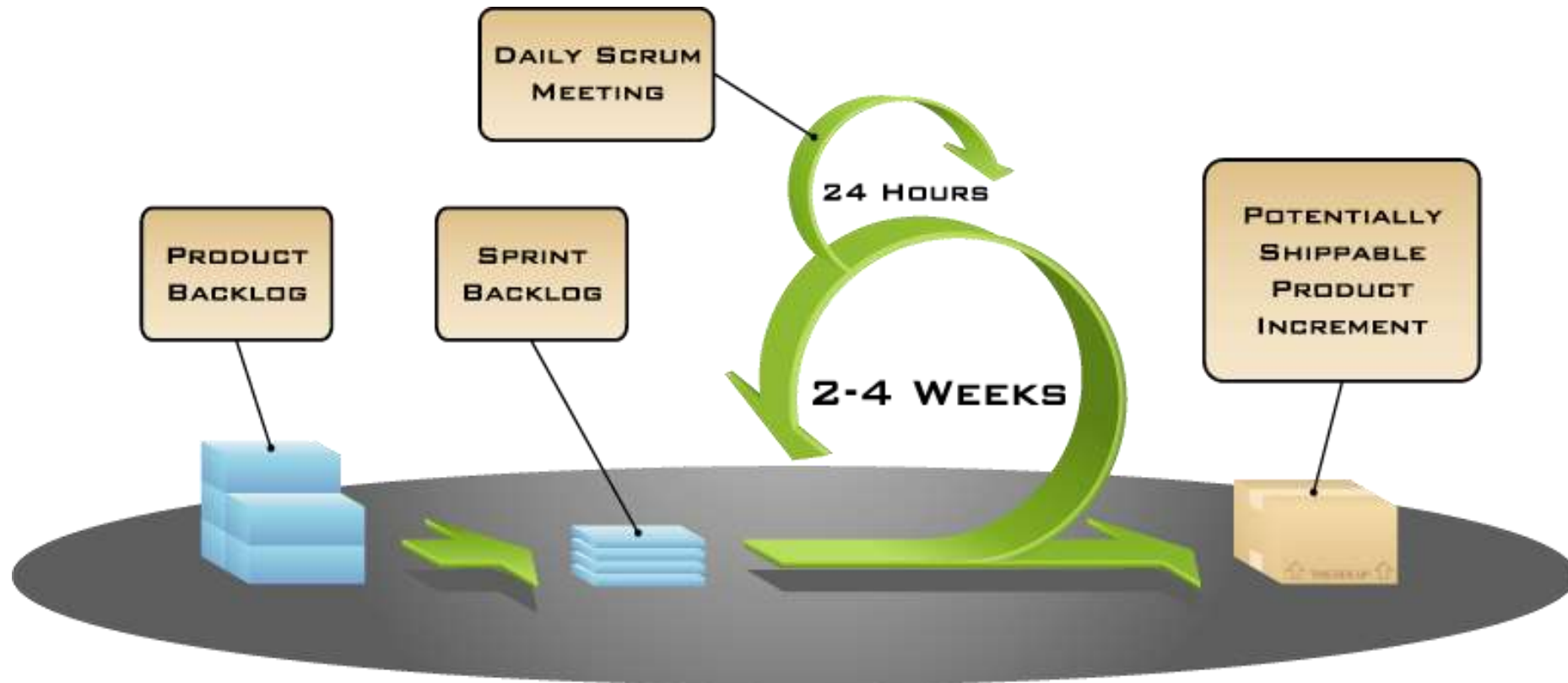


Source: www.agilemanifesto.org

Scrum



Scrum



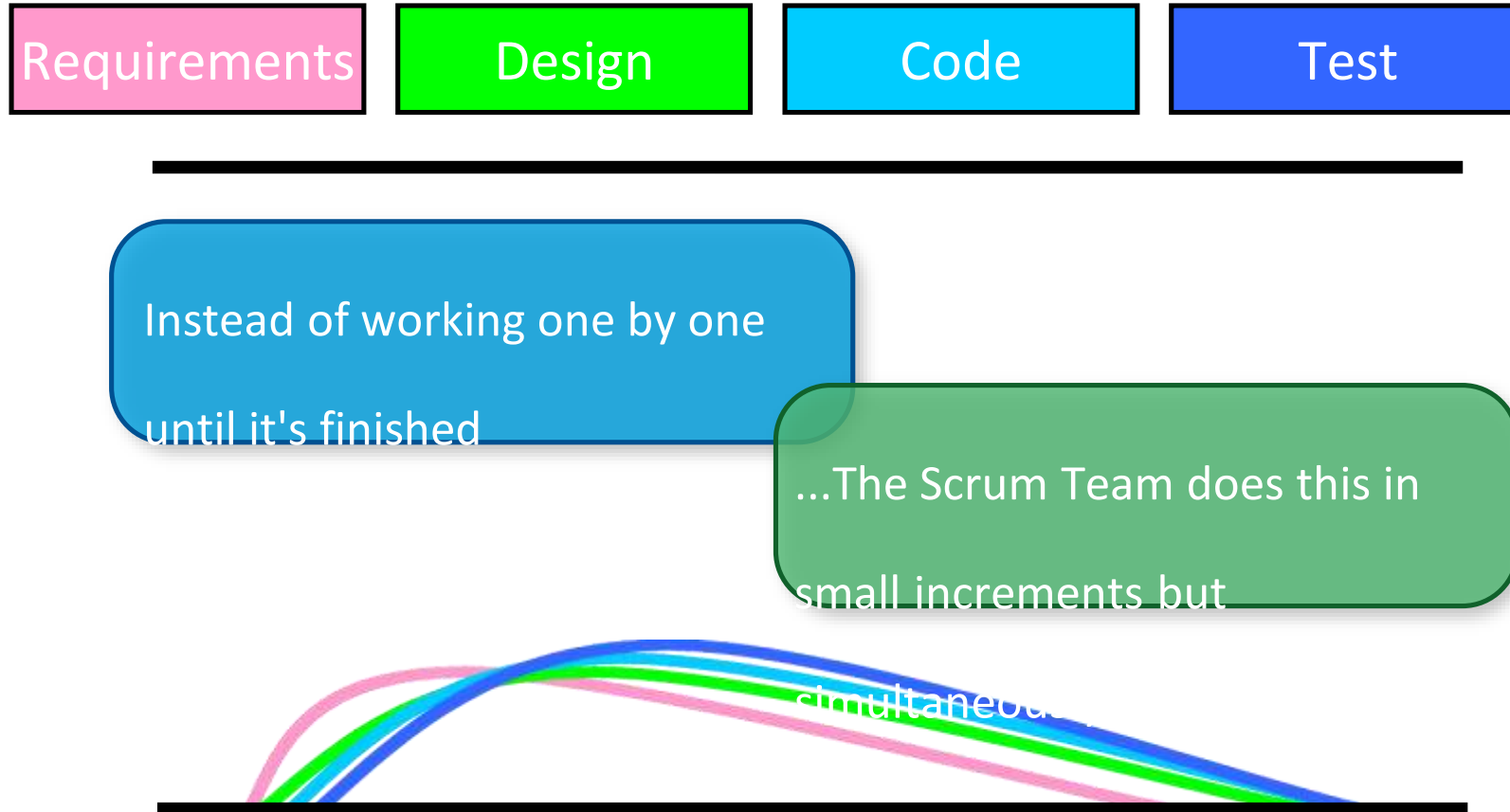
COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Source: www.mountaingoatsoftware.com/scrum

Sprints

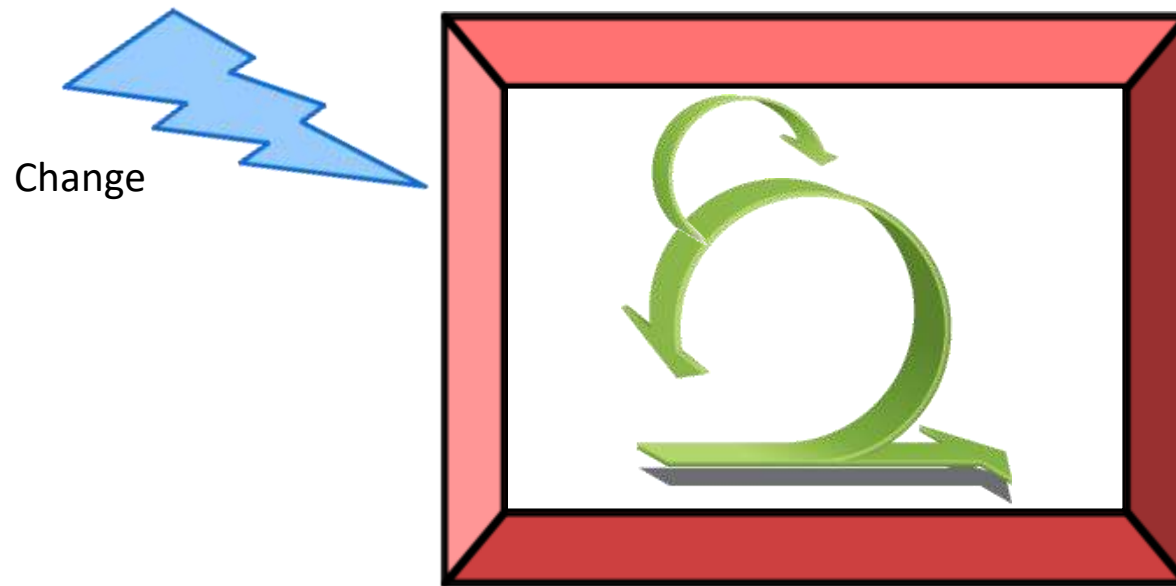
- Progress on a Scrum project is carried out in “Sprints”.
 - Or known as "iteration" in the Extreme Programming method.
- The general duration is 2-4 weeks or a maximum of 1 month.
- Consistent duration will produce a good team work rhythm.
- Software is designed, coded, and tested in each Sprint.

Sequential vs. Sequential overlapping development



Source: "The New New Product Development Game" by Takeuchi and Nonaka. *Harvard Business Review*, January 1986.

There should be no changes while the Sprint is running..



- The duration of the Sprint must be planned according to the team's ability to prevent changes from occurring within one Sprint.

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

Events

- Sprint planning
- Sprint review
- Sprint retrospectives
- Daily scrum meetings

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team Member

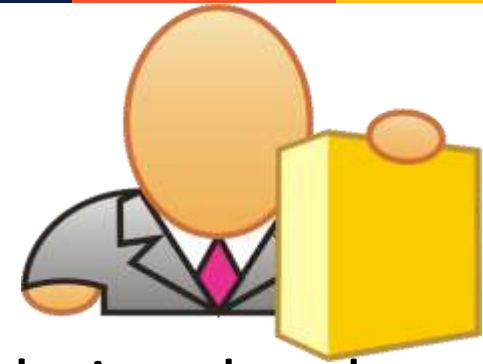
Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospectives
- Daily scrum meetings

Artifacts

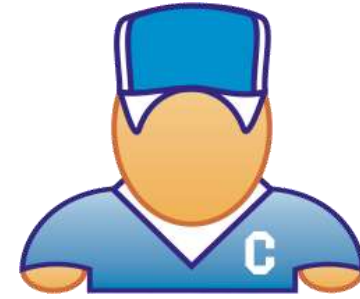
- Product backlog
- Sprint backlog
- Burndown charts

Product owner



- Determine what features must be in the product being developed.
- Deciding content and release time.
- Responsible for product profits (ROI).
- Determine feature priorities based on market conditions.
- Reorganize features and their priorities at each iteration, as needed.
- Accept and/or reject team work results.

The ScrumMaster



- Play a “management” role in the project.
- Responsible for upholding Scrum values and practices.
- Remove barriers/obstacles.
- Ensure the team is fully functional and productive.
- Ensure cooperation and closeness to all *roles* and function within the team.
- Protect the team from external interference.



Team Member

- Generally consists of 5-9 people.
- *Cross-functional*:
 - Programmer, QA Engineer, User, UI/UX Designer, etc.
- All members must work full time.
 - Although sometimes there are exceptions (Example: database administrator)
- Teams are independent, *self-organizing*
 - Ideally there is no position/title, but sometimes (very rarely) there can be.
- Membership may not change while the Sprint is in progress.
 - Can be replaced when there is a break between two Sprints.

Scrum Framework

Roles

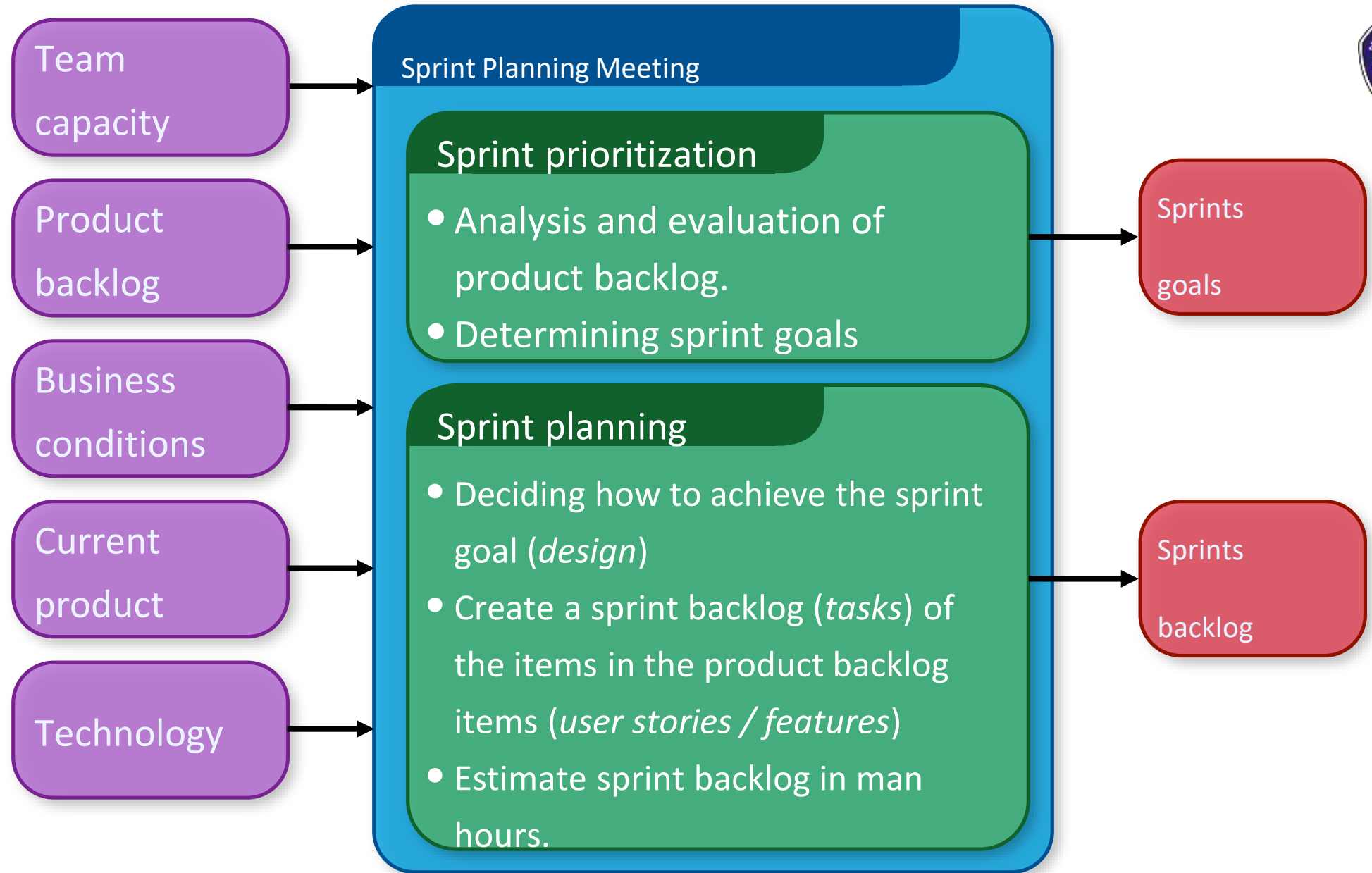
- Product owner
- ScrumMaster
- Team

Events

- Sprint planning
- Sprint review
- Sprint retrospectives
- Daily scrum meetings

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts



Sprint planning

- The team selects items from the product backlog that they are confident they can complete within 1 Sprint.
- Creating a Sprint Backlog:
 - Each existing task is identified and estimated how long it will take in hours (1-16 hours)
 - Done collaboratively, not alone by the Scrum Master.
- The consideration used is "high level design" or "user story".

As someone who is going on holiday, I want to see photos of the available hotels.



Middle tier coding (8 hours)

Working on the user interface (4 hours)

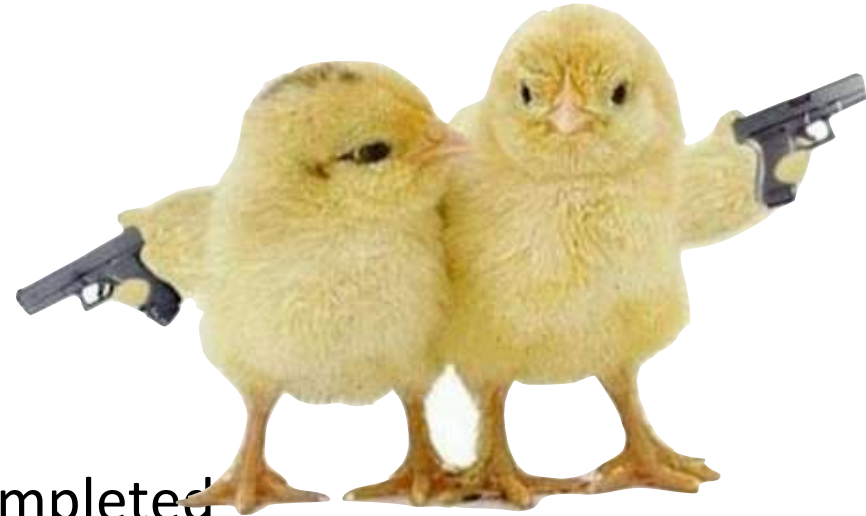
Making test fixtures (4 hours)

Creating the main classes (6 hours)

Updating performance tests (4 hours)

Event #1: Daily scrum

- Characteristic:
 - Daily
 - A maximum of 15 minutes must be completed
 - Stand up (No, seriously. The one whose turn it is to speak cannot sit down)
- Not to “solve problems”
 - Everyone is invited.
 - But only those who are allowed to speak are: Team Members, ScrumMaster, and Product Owner
- Helps to avoid prolonged meetings and going here and there.



In Daily Scrums, everyone shares:

1
What were you doing
yesterday?

2
What are you going to do
today?

3
Is it available *blockers*?

- Here all team members must be honest. It's not just an image if it looks like you were working yesterday even though you were just doing it *youtube-an*.
 - Each person is committed to all his colleagues.

Event #2: Sprint Review

- The team presents what has been achieved at the end of one recently completed Sprint.
- Generally in the form of demonstrating new features of the application being created or explaining the architecture in the application.
- Informal
 - Maximum 4 hours must be completed.
 - Don't use PPT.
- All teams participate.
- Invite all related people and/or parties.



Event #4: Sprint retrospective

- Periodically review what is working and what is not working well.
- Maximum 3 hours if the sprint is 1 month, or shorter.
- Done every time a Sprint is finished.
- Everyone takes part:
 - ScrumMaster
 - Product owner
 - Team Member
 - It could be our clients and others who need/are interested.

In the Sprint Retrospective, we discuss:

Start/Stop/Continue

- The whole team gets together and discusses what they want:

Starting to do it

Stop doing

Keep doing it

Not only technically related, but it could be anything. As long as everyone agrees.
Example: "Mom, we need snacks"

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

Events

- Sprint planning
- Sprint review
- Sprint retrospectives
- Daily scrum meetings

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Product backlog



This is the
product backlog

- System Requirements.
- A list of work to be achieved on a project.
- Ideally expressed such that each item has *value* which is real for *end-user*.
- Priorities are set by the Product Owner.
- Priorities can be reset at the start of each sprint.

Example of a Product Backlog

Backlog Items	Estimate
Allow a guest to make a reservation. Features for guests to make reservations.	3
As a guest, I would like to cancel the booking.	5
As a guest, I would like to change the booking dates.	3
As a hotel employee I want to be able to make income reports per room.	8
Repair <i>errorAndexception</i> .	8
...	30
...	50



Sprint Goals

- A short statement that emphasizes where our work will focus during a Sprint.

Database Application

Making our application run on SQL Server and Oracle

Life Sciences

Provides the features needed for population genetic studies.

Fintech Applications

Helps provide technical indicators in real-time with streaming data.

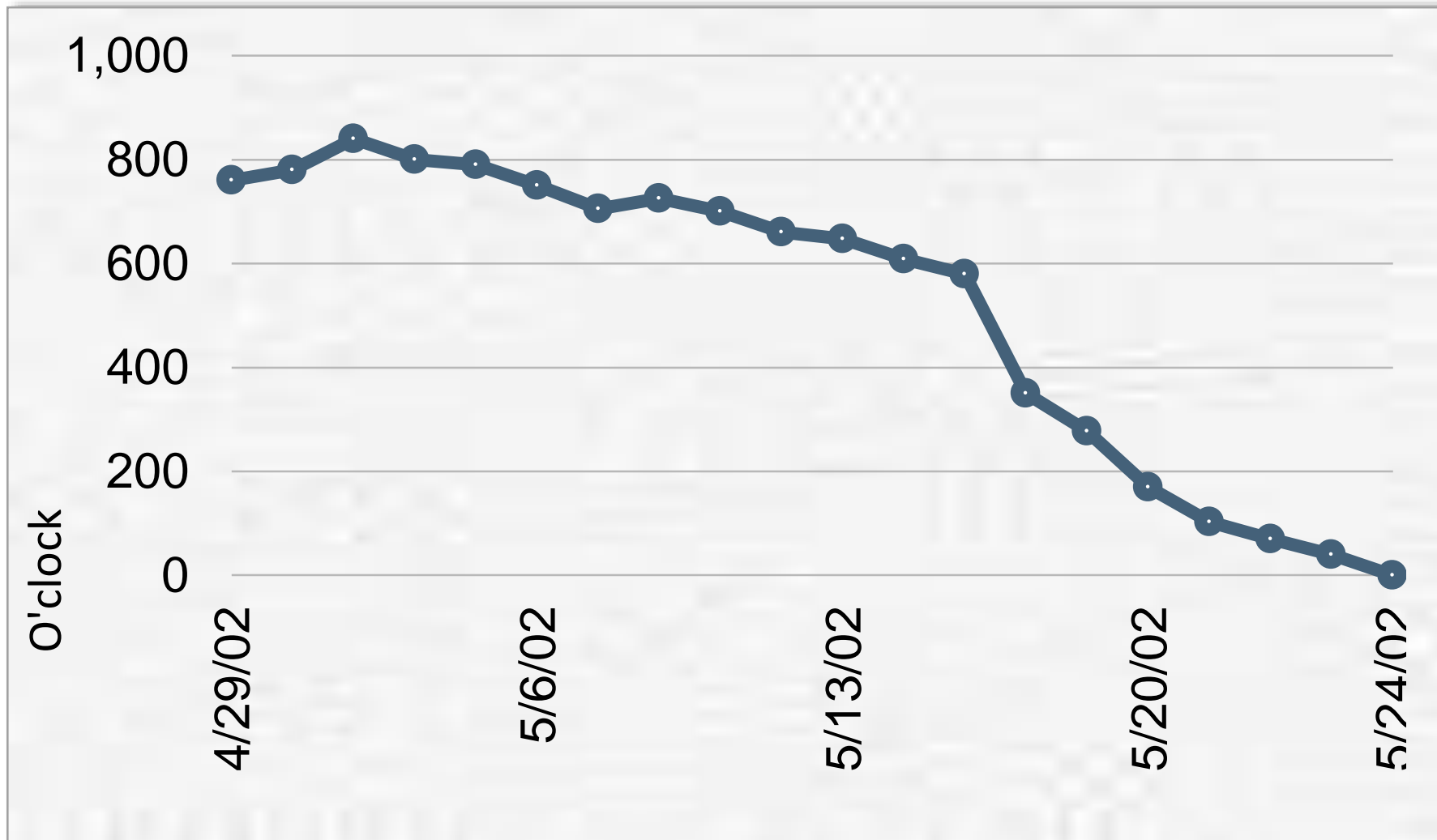
Managing the Sprint Backlog

- Each person chooses for himself which backlog he wants to work on.
 - No assignments.
- Estimates of outstanding work are updated daily.
- Any team member may add, delete, or change the Sprint Backlog.
- Start and keep working for the duration of the Sprint.
- If there is work that is unclear, make it into one Sprint Backlog item with an increased time estimate to be broken down later.
- Update remaining unfinished work as the overall work becomes clearer.

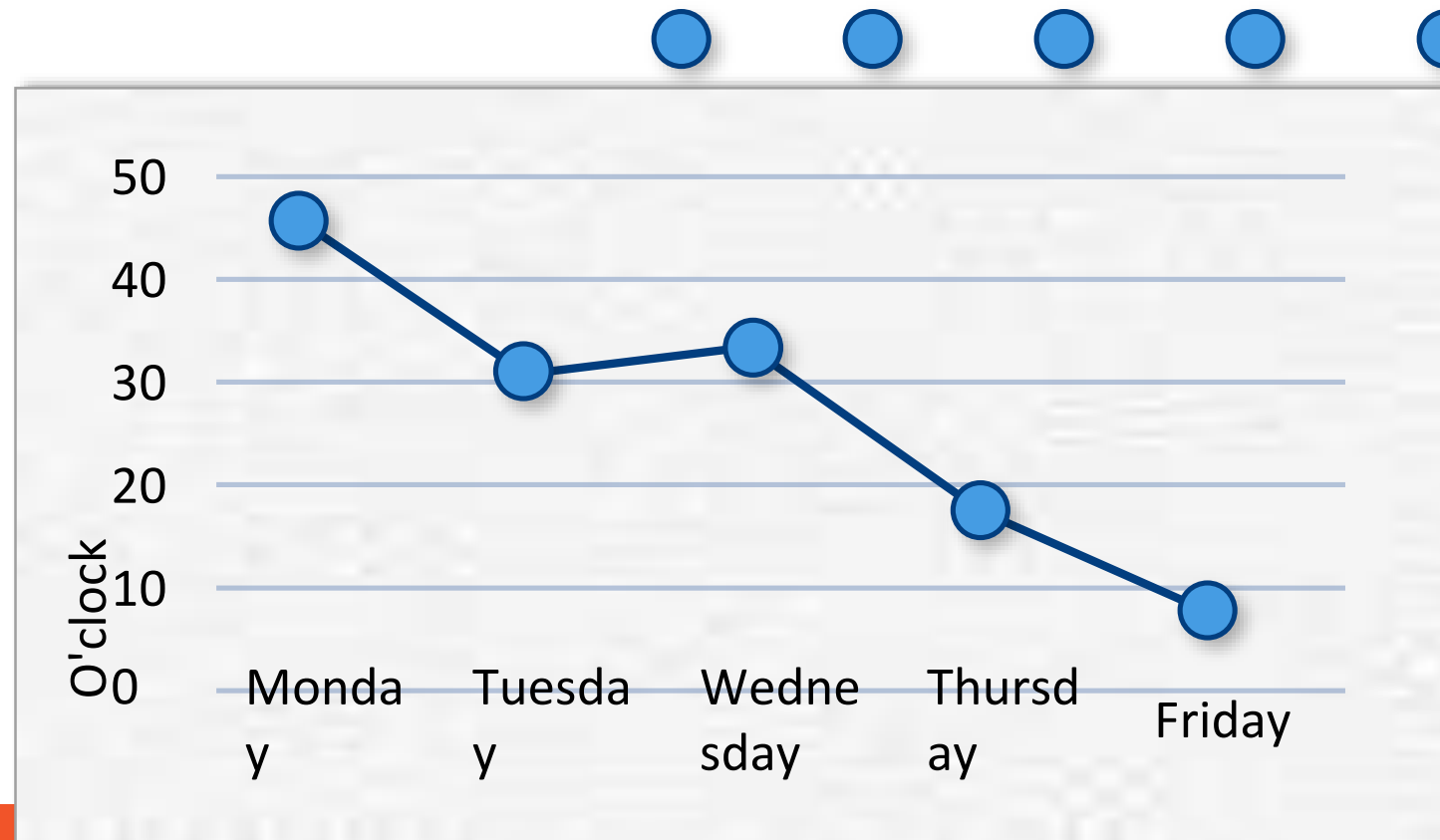
Example of a Sprint Backlog

Work	Monday	Tuesday	Wednesday	Thursday	Friday
UI Coding	8	4	8		
Middleware coding	16	12	10	4	
Test middleware	8	16	16	11	8
Create a help menu	12				
Create main classes	8	8	8	8	8
Add error log			8	4	

Example of a “Sprint Burndown” Diagram



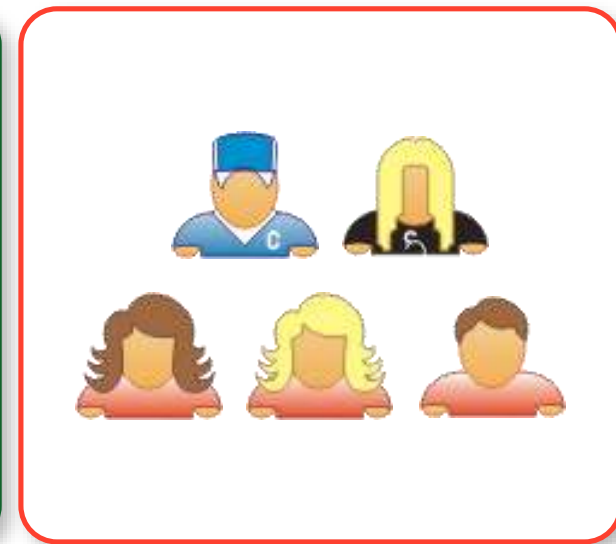
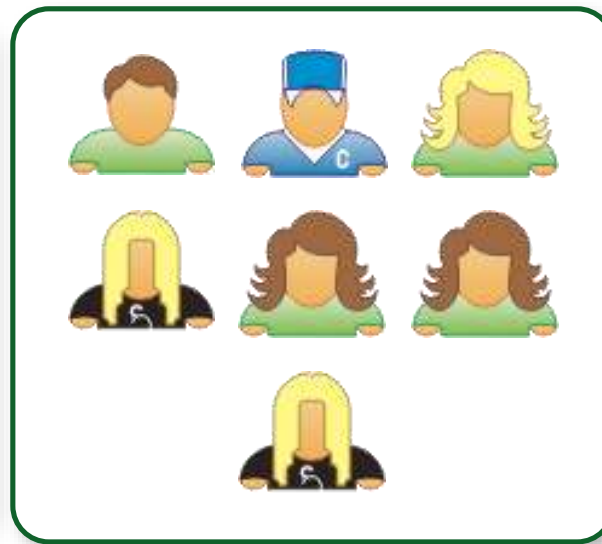
Work	Sen	Cell	Wed	Thu	Fri
UI Coding	8	4	8		
Middleware coding	16	12	10	7	
Testing middleware	8	16	16	11	8
Create a Help menu	12				



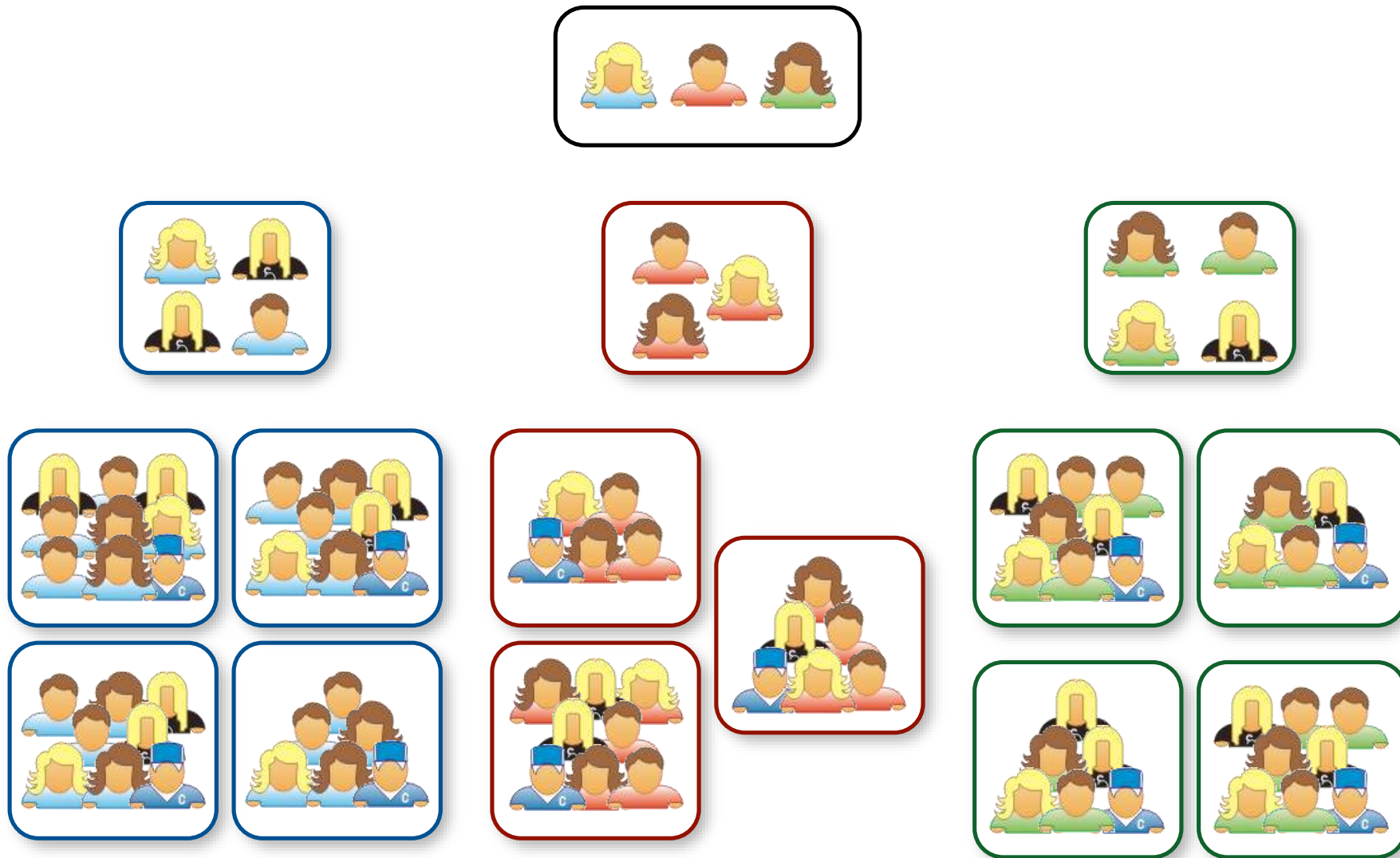
Scalability

- Generally each team consists of 7 ± 2 people
 - Scalability can be achieved with a “team within a team” setup
- Scalability factor
 - The type of application being worked on
 - Team size
 - Team distribution
 - Project duration
- Scrum is proven, and can be used in projects of 500+ people.

Scaling through the Scrum of scrums



Scrum of scrums of scrums





Any questions?