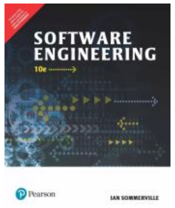
(SCS1207) SOFTWARE ENGINEERING

Dr. Lasanthi. N. C De Silva

lnc@ucsc.cmb.ac.lk

Create a WhatsApp Group
0773889156

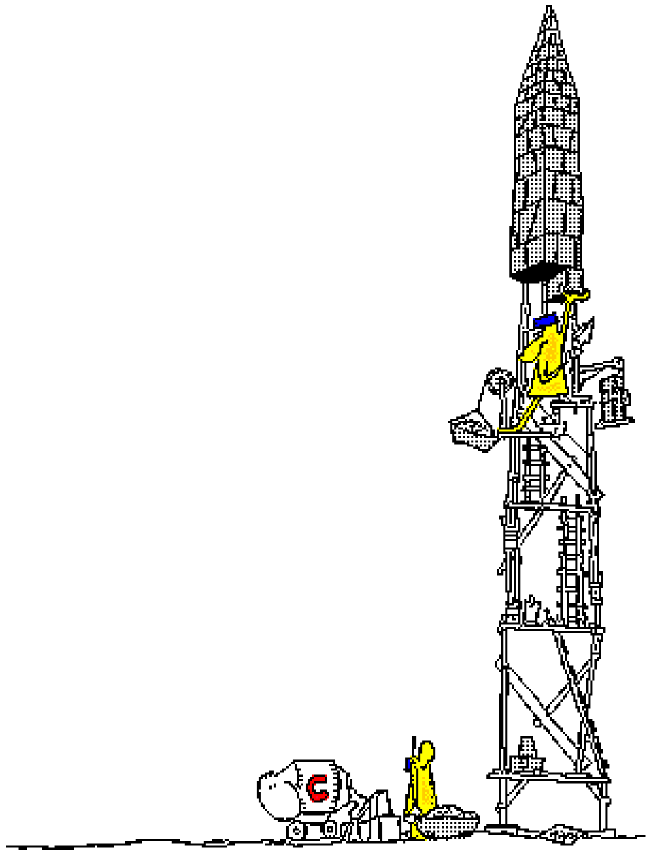
RECOMMENDED READING



Main Reference:

- Ian Sommerville, Software Engineering, 10th Edition, Pearson, 2017.
- R Pressman, Software Engineering - A Practitioners Approach, 7th Edition, McGraw Hill.

COURSE CONTENT

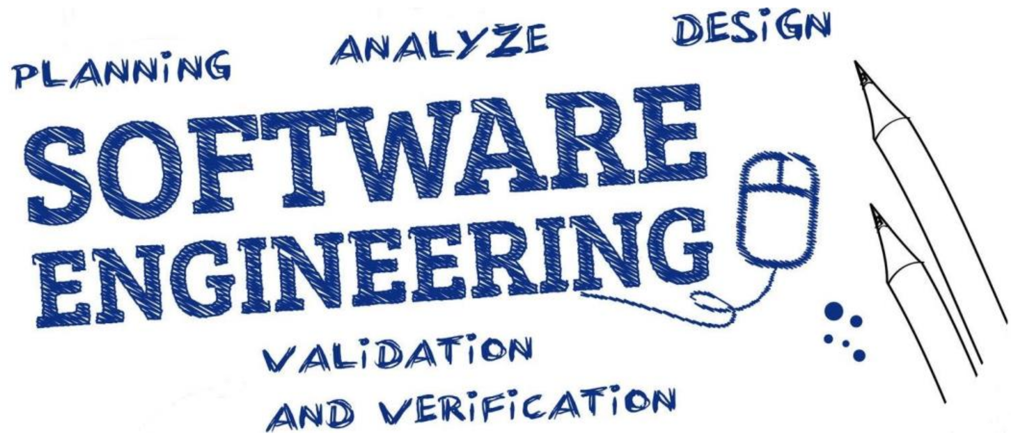


Chapter 1	Introduction
Chapter 2	Software Process Models
Chapter 3	Agile Software Development
Chapter 4	Requirements Engineering
Chapter 5	System Modeling
Chapter 6	Architectural Design
Chapter 7	Design and Implementation
Chapter 8	Software Testing
Chapter 9	Software Evolution

RECAP: CHAPTER 1

What is software engineering?

What is the difference between SWE and CS?



WHY NEED SOFTWARE ENGINEERING?

RECAP: CHAPTER 2

- What is a software process?
- What is a software process model?
- How many software process models are there?
- Name few software process models?
- What are the fundamental software engineering activities?

Software Specification

Software Validation

Software Development

Software Evolution

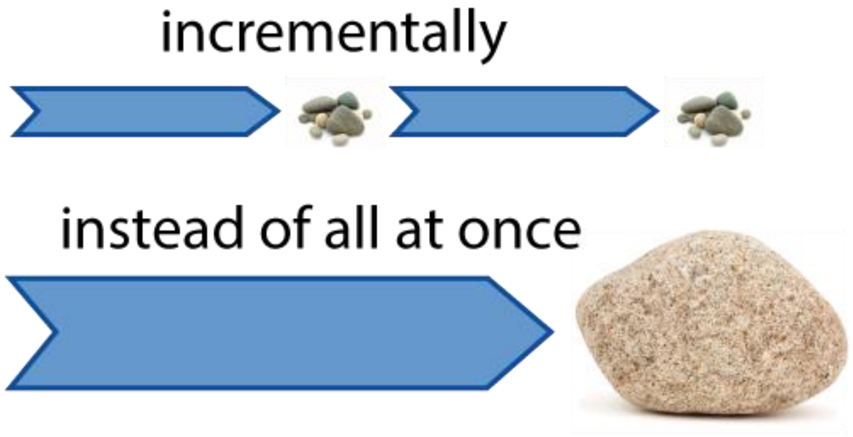
Chapter 3
Agile Software Development

Learning Outcomes

- ✧ Define agile software development
- ✧ Understand the rationale for agile software development methods and agile manifesto.
- ✧ Compare and contrast agile and plan-driven development.
- ✧ Identify the key practices in extreme programming and how these relate to general practices of agile methods.
- ✧ Understand Scrum, Kanban, Lean approach to agile software development.

An Introduction to Agile?

- Why does it exist?
- What problems does it solve?
- What is it?



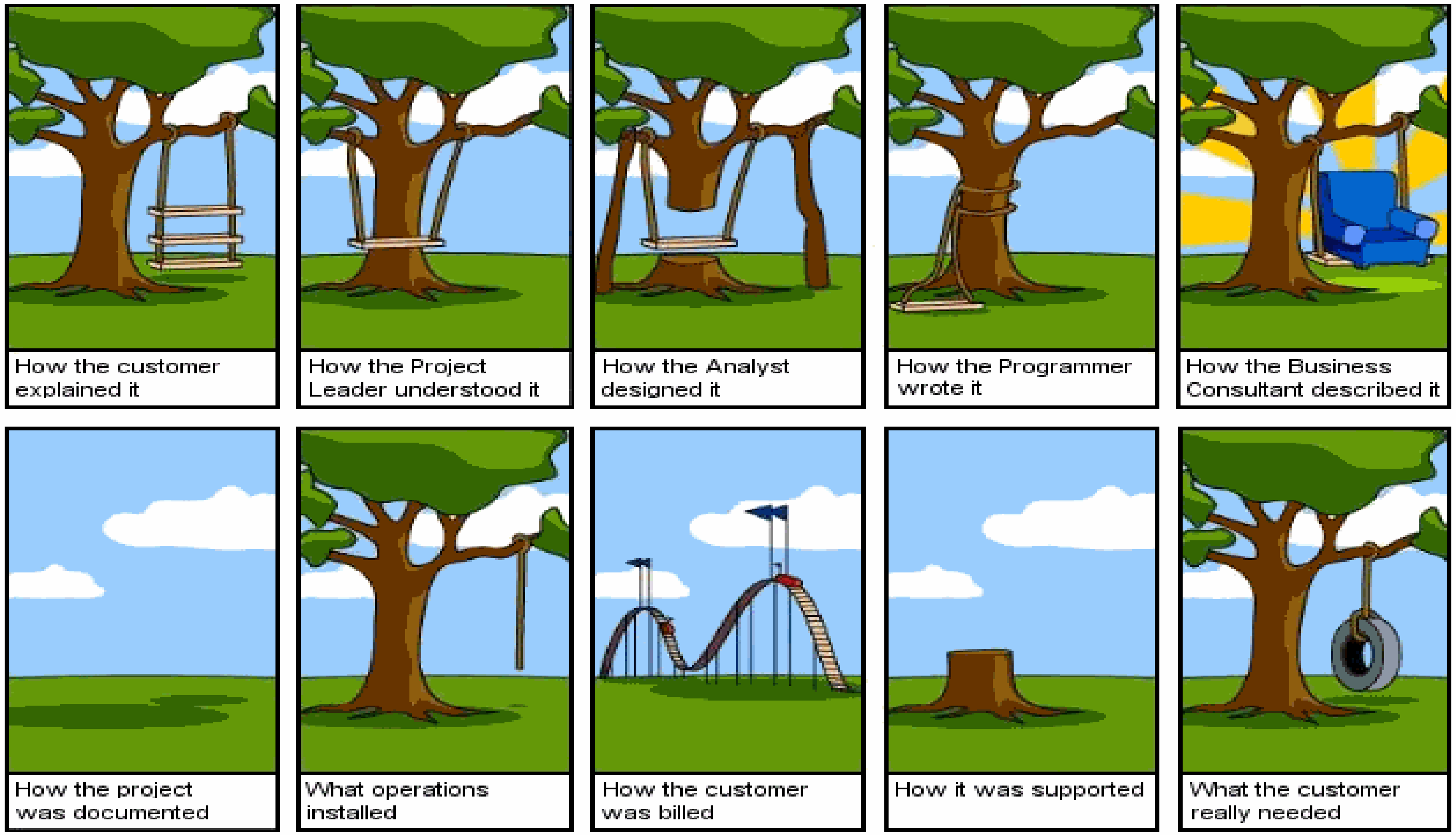
- Agile is a **time boxed**, **iterative approach** to software delivery that builds software **incrementally** from the start of the project, instead of trying to deliver it all at once near the end.

Most software developments fail

- ✧ Software developments projects being **cancelled** every now and then
- ✧ Software projects are being considered as **failures** by those who initiated it.
- ✧ One in every two projects **exceed its budget** by 200%.

WHY?

WHY / What is wrong?


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

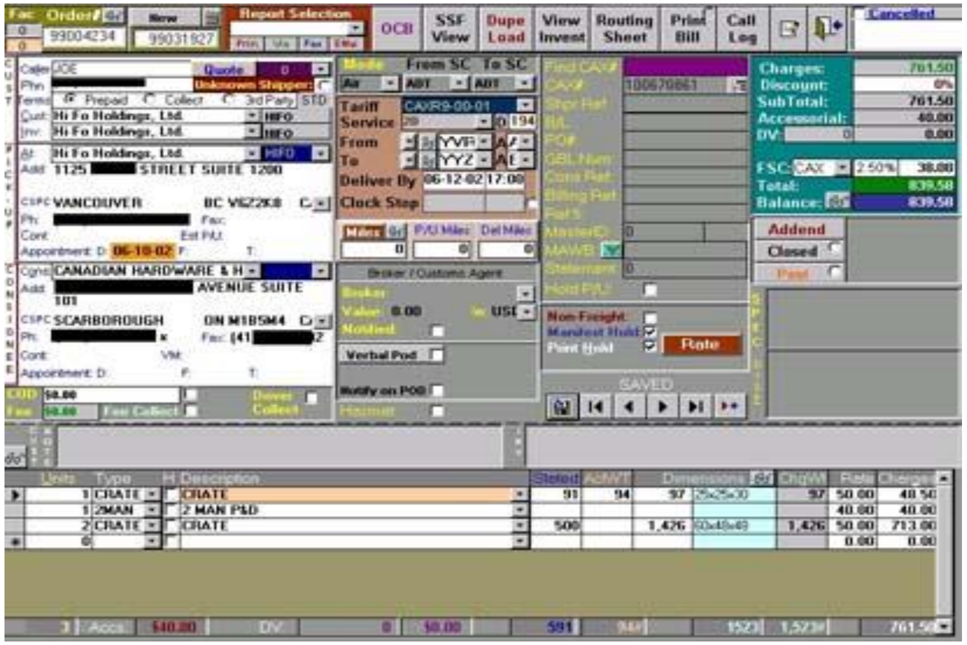
WHY? Cont.…

✧ Do not meet the need of the user.

✧ Deadline rush.

✧ Less number of features delivered.

✧ Poor interfaces.



Order# 99004234 New 99031927

Caller JOE Quote 0
Phn [redacted] Unknown Shipper:
Terms (● Prepaid ○ Collect ○ 3rd Party) STD
Cust: Hi Fo Holdings, Ltd. HIFO
Inv: Hi Fo Holdings, Ltd. HIFO
At: Hi Fo Holdings, Ltd. HIFO
Add: 1125 [redacted] STREET SUITE 1200
CSPC VANCOUVER BC V6Z2K8 C
Ph: [redacted] Fax:
Cont: Est P/U:
Appointment: D: 06-10-02 F: T:

Cgne CANADIAN HARDWARE & H
Add: [redacted] AVENUE SUITE 101
CSPC SCARBOROUGH ON M1B5M4 C
Ph: [redacted] x Fax: (41[redacted]2
Cont: VM:
Appointment: D: F: T:

COD $0.00 Driver Collect
Fee $0.00 Fee Collect

Mode: Air From SC: ABT To SC: ABT
Tariff CAXR9-00-01
Service 29 194
From YVR
To YYZ
Deliver By 06-12-02 17:00
Clock Stop
Miles 0 P/U Miles 0 Del Miles 0

Broker / Customs Agent
Broker
Value 0.00 in USD
Notified
Verbal Pod
Notify on POD
Hazmat

Find CAX#
CAX# 100670861
Shpr Ref
B/L
PO#
GBL Num
Cons Ref
Billing Ref
Ref 5
MasterID 0
MAWB
Statement 0
Hold P/U
Non Freight
Manifest Hold ✓
Print Hold ✓

Charges: 761.50
Discount: 0%
SubTotal: 761.50
Accessorial: 40.00
DV: 0 0.00
FSC: CAX 2.50% 38.08
Total: 839.58
Balance: 839.58

Units	Type	H	Description	Stated	AdWT	Dimensions		ChgWt	Rate	Charges
1	CRATE		CRATE	91	94	97	25x25x30	97	50.00	48.50
1	2MAN		2 MAN P&D						40.00	40.00
2	CRATE		CRATE	500		1,426	60x48x49	1,426	50.00	713.00
0									0.00	0.00

3 Accs: $40.00 DV: 0 $0.00 591 94# 1523 1,523# 761.50

SUCCESS IS NOT JUST FUNCTIONS ANY MORE!!!

LOT more EXPECTATIONS - Users

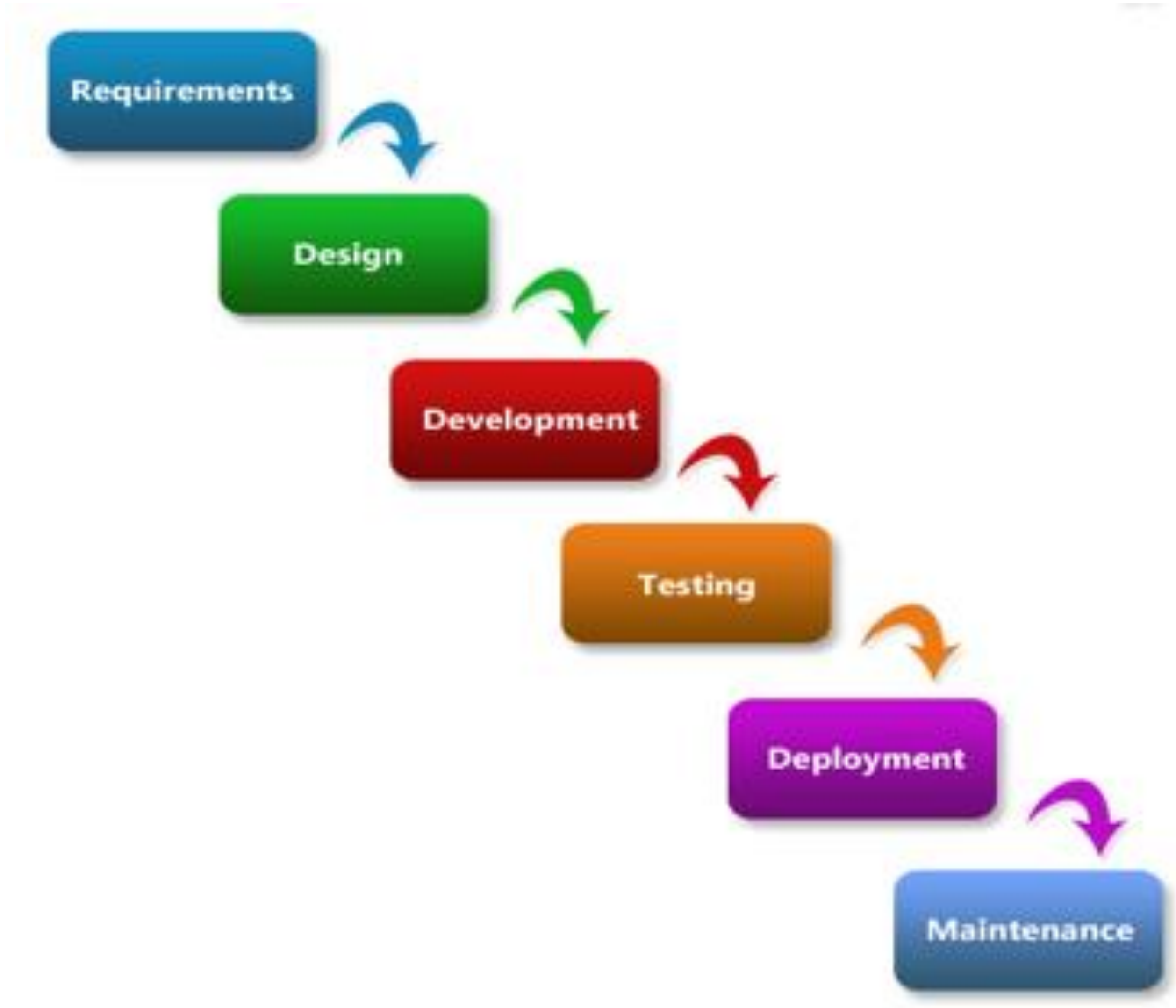
✧ Speed Delivery

✧ Accuracy

✧ Features vs Usability

WHAT CAUSES MOST OF THE SOFTWARE DEVELOPMENTS FAIL?????????

Traditional Waterfall Method



Waterfall model build-upon Assumptions

- ✧ Requirements can be entirely predicted upfront.
- ✧ Each phase of the lifecycle can be perfected before moving to the next.
- ✧ Timeframes and budgets can be predicted up front.
- ✧ The feedback from the real user is not so valuable.

UNREALISTIC

Causes for Failures

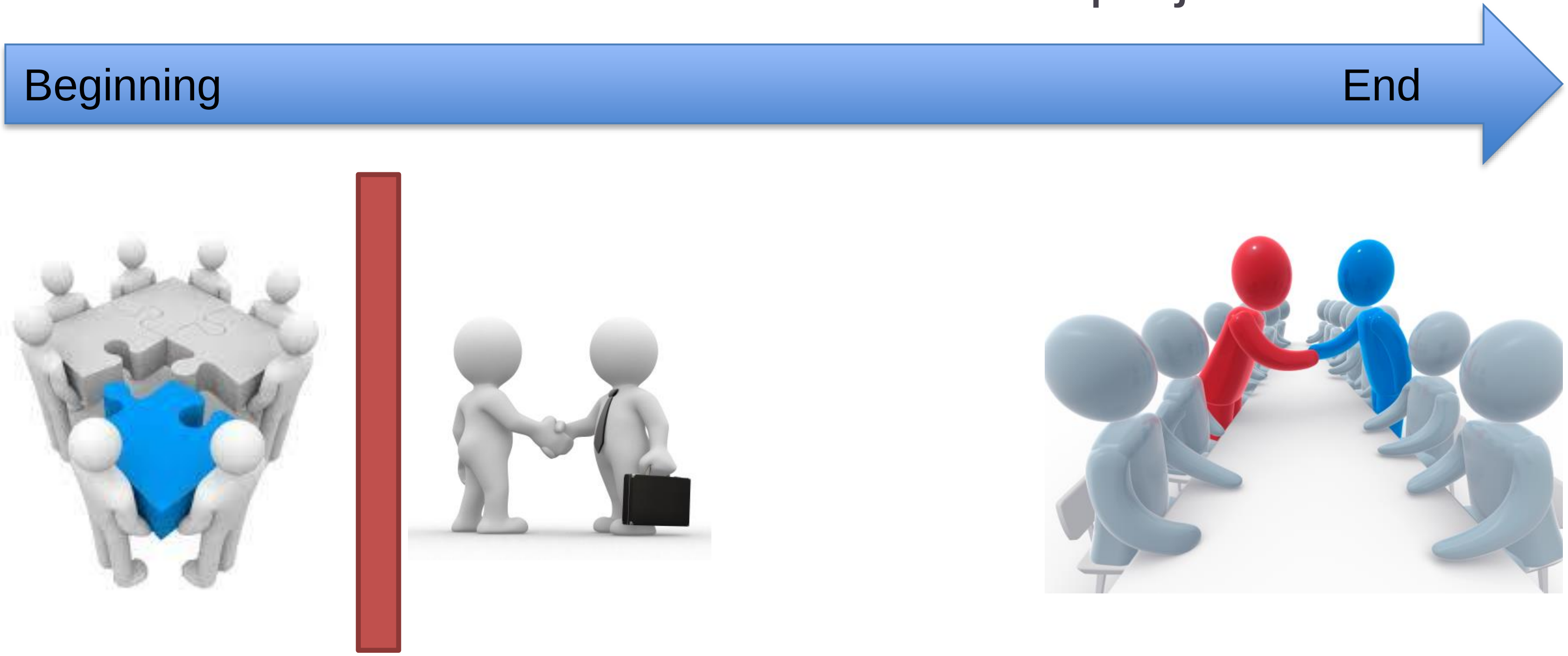
✧ Risk is bundled!

Functions	Functions	Functions	Functions
Functions	Functions	Functions	Functions
Functions	Functions	Functions	Functions
Functions	Functions	Functions	Functions

Causes for Failures

✧ Poor communication!

✧ Feedback is obtained at the end of the project



Can you meet the client requirement at this stage????

AGILE METHODS

Rapid software development

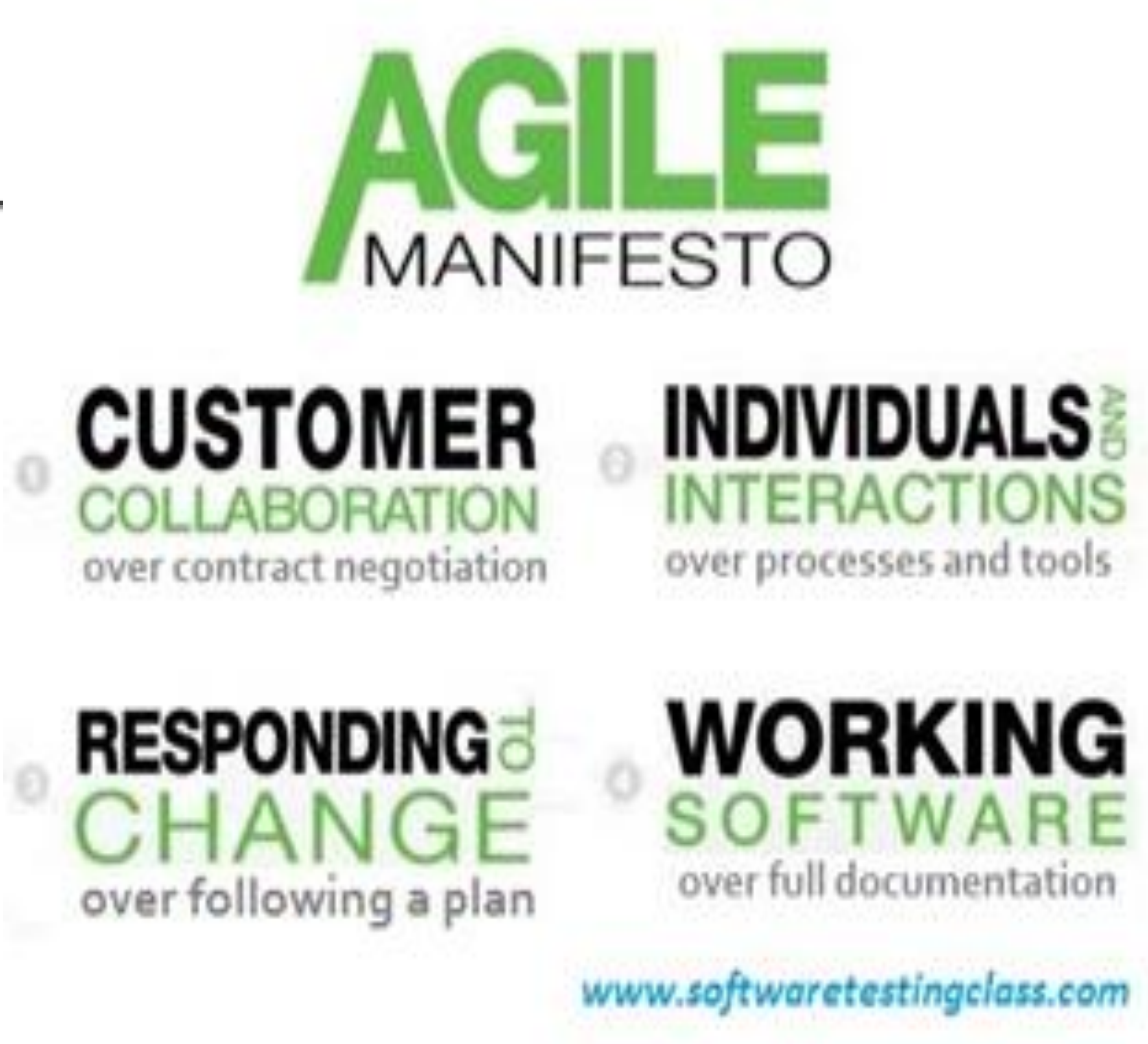
- ✧ Rapid development and delivery is now often the most important requirement for software systems
  - Businesses operate in a fast –changing requirement and it is practically impossible to produce a set of stable software requirements
  - Software must evolve quickly to reflect changing business needs.
- ✧ Plan-driven development is essential for some types of system but does not meet these business needs.
- ✧ Agile development methods emerged in the late 1990s whose aim was to radically reduce the delivery time for working software systems

Agile methods

✧ Dissatisfaction with the overheads involved in software design methods of the 1980s and 1990s led to the creation of agile methods. These methods:

- Focus on the **code** rather than the design
- Are based on an **iterative approach** to software development
- Are intended to **deliver working software quickly** and evolve this quickly to meet changing requirements.

✧ The aim of agile methods is to reduce overheads in the software process (e.g. by limiting documentation) and to be able to respond quickly to changing requirements without excessive rework.



AGILE MANIFESTO

CUSTOMER COLLABORATION over contract negotiation

INDIVIDUALS AND INTERACTIONS over processes and tools

RESPONDING TO CHANGE over following a plan

WORKING SOFTWARE over full documentation

www.softwaretestingclass.com

Agile manifesto cont.…

- *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.*

12 Agile Principles

1.	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	7.	Working software is the primary measure of progress.
2.	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	8.	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
3.	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	9.	Continuous attention to technical excellence and good design enhances agility.
4.	Business people and developers must work together daily throughout the project.	10.	Simplicity--the art of maximizing the amount of work not done--is essential.
5.	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	11.	The best architectures, requirements, and designs emerge from self-organizing teams.
6.	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	12.	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

12 Agile Principles

1.Early and continuous delivery of valuable software

1.	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	7.	Working software is the primary measure of progress.
			ce indefinitely.
3.	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	9.	Continuous attention to technical excellence and good design enhances agility.
4.	Business people and developers must work together daily throughout the project.	10.	Simplicity--the art of maximizing the amount of work not done--is essential.
5.	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	11.	The best architectures, requirements, and designs emerge from self-organizing teams.
6.	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	12.	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

12 Agile Principles

1.	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	7.	Working software is the primary measure of progress.
2.	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	8.	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
	preference to the shorter timescale.		ttention to technical excellence and enhances agility.
4.	Business people and developers must work together daily throughout the project.	10.	Simplicity--the art of maximizing the amount of work not done--is essential.
5.	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	11.	The best architectures, requirements, and designs emerge from self-organizing teams.
6.	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	12.	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

2. Welcome Changing Requirements

12 Agile Principles

1.	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	7.	Working software is the primary measure of progress.
2.	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	8.	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
3.	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	9.	Continuous attention to technical excellence and good design enhances agility.
			maximizing the amount of sential.
	them the environment and support they need, and trust them to get the job done.		es, requirements, and designs emerge from self-organizing teams.
6.	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	12.	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

3. Deliver working software frequently

12 Agile Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	7. Working software is the primary measure of progress.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	9. Continuous attention to technical excellence and good design enhances agility.
4. Business people and developers must work together daily throughout the project.	10. Simplicity--the art of maximizing the amount of work not done--is essential.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

4. Business people and developers must work together

12 Agile Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	7. Working software is the primary measure of progress.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	9. Continuous attention to technical excellence and good design enhances agility.
4. Business people and developers must work together daily throughout the project.	10. Simplicity--the art of maximizing the amount of work not done--is essential.
5. Build projects around motivated individuals. Give them the environment and support their need, and trust them to get the job done.	11. The best architectures, requirements, and designs emerge from self-organizing teams.
	s on how to nd adjusts

5. Build projects around motivated individuals

12 Agile Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.

6. Face-to-face conversation

and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

12 Agile Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	7. Working software is the primary measure of progress.
2. Welcome c developme the custome[r's competitive advantage.]	users should be able to maintain a constant pace indefinitely.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	9. Continuous attention to technical excellence and good design enhances agility.
4. Business people and developers must work together daily throughout the project.	10. Simplicity--the art of maximizing the amount of work not done--is essential.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	11. The best architectures, requirements, and designs emerge from self-organizing teams.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

7. Working software to measure the progress

12 Agile Principles

1.	**Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.**	**7.**	**Working software is the primary measure of progress.**
2.	**Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.**	**8.**	**Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.**
3.	**Deliver working software frequ couple of weeks to a couple of preference to the shorter time**		
4.	**Business people and developers must work together daily throughout the project.**	**10.**	**Simplicity--the art of maximizing the amount of work not done--is essential.**
5.	**Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.**	**11.**	**The best architectures, requirements, and designs emerge from self-organizing teams.**
6.	**The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.**	**12.**	**At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.**

8. Promote sustainable development

12 Agile Principles

1.	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	7.	Working software is the primary measure of progress.
2.	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	8.	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
3.	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	9.	Continuous attention to technical excellence and good design enhances agility.
4.	Business people and d together daily through		
5.	Build projects around them the environment and support they need, and trust them to get the job done.		designs emerge from self-organizing teams.
6.	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	12.	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

9. Technical excellence and good design.

12 Agile Principles

1.	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	7.	Working software is the primary measure of progress.
2.	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	8.	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
3.	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	9.	Continuous attention to technical excellence and good design enhances agility.
4.	Business people and developers must work together daily throughout the project.	10.	Simplicity--the art of maximizing the amount of work not done--is essential.
5.	Build proje them the e and trust tl		
6.	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	12.	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

10. Maximizing the amount of work not done.

12 Agile Principles

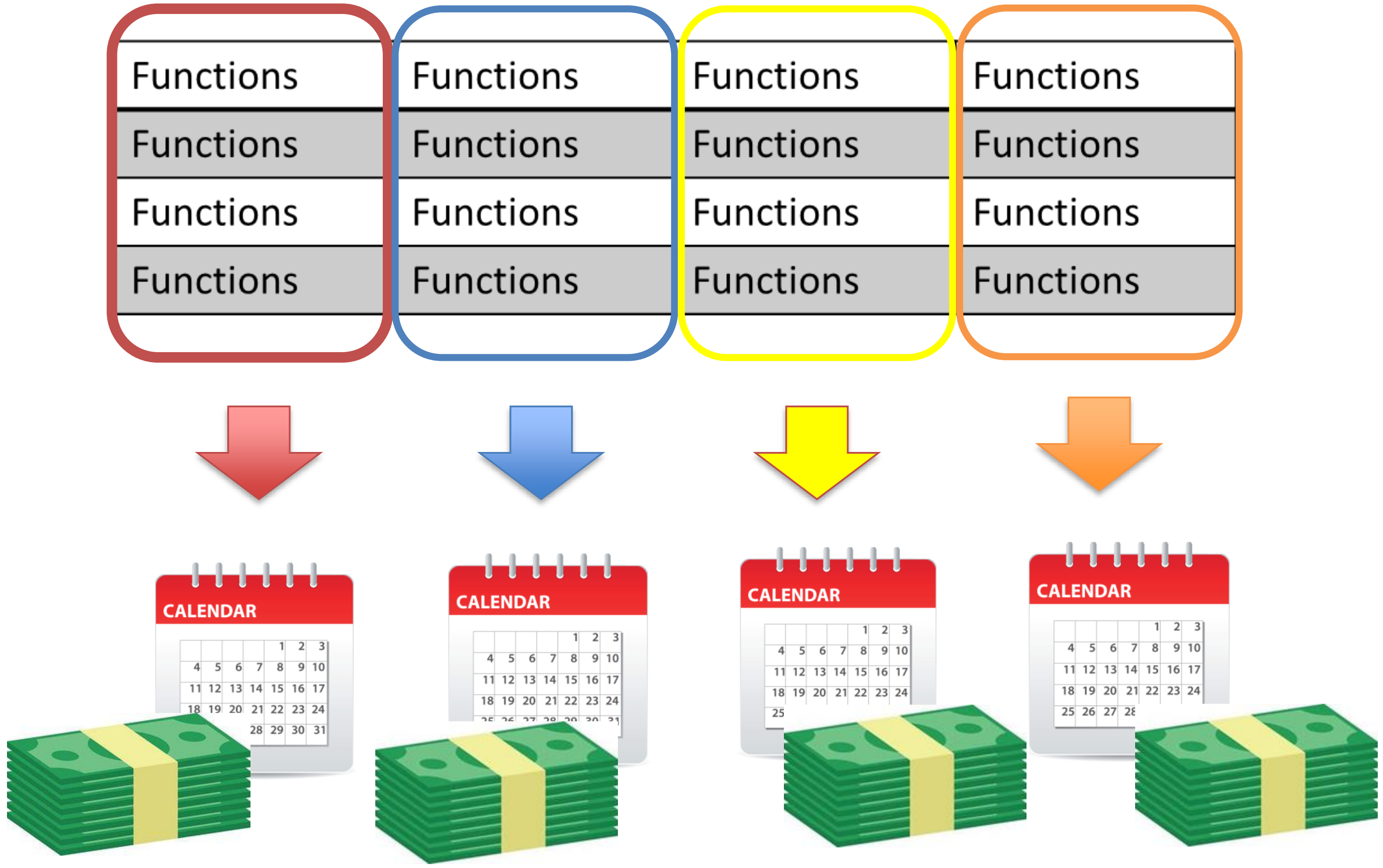
1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	7. Working software is the primary measure of progress.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	9. Continuous attention to technical excellence and good design enhances agility.
4. Business people and developers must work together daily throughout the project.	10. Simplicity--the art of maximizing the amount of work not done--is essential.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	11. The best architectures, requirements, and designs emerge from self-organizing teams.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	its behavior accordingly.

11. Self-organizing teams.

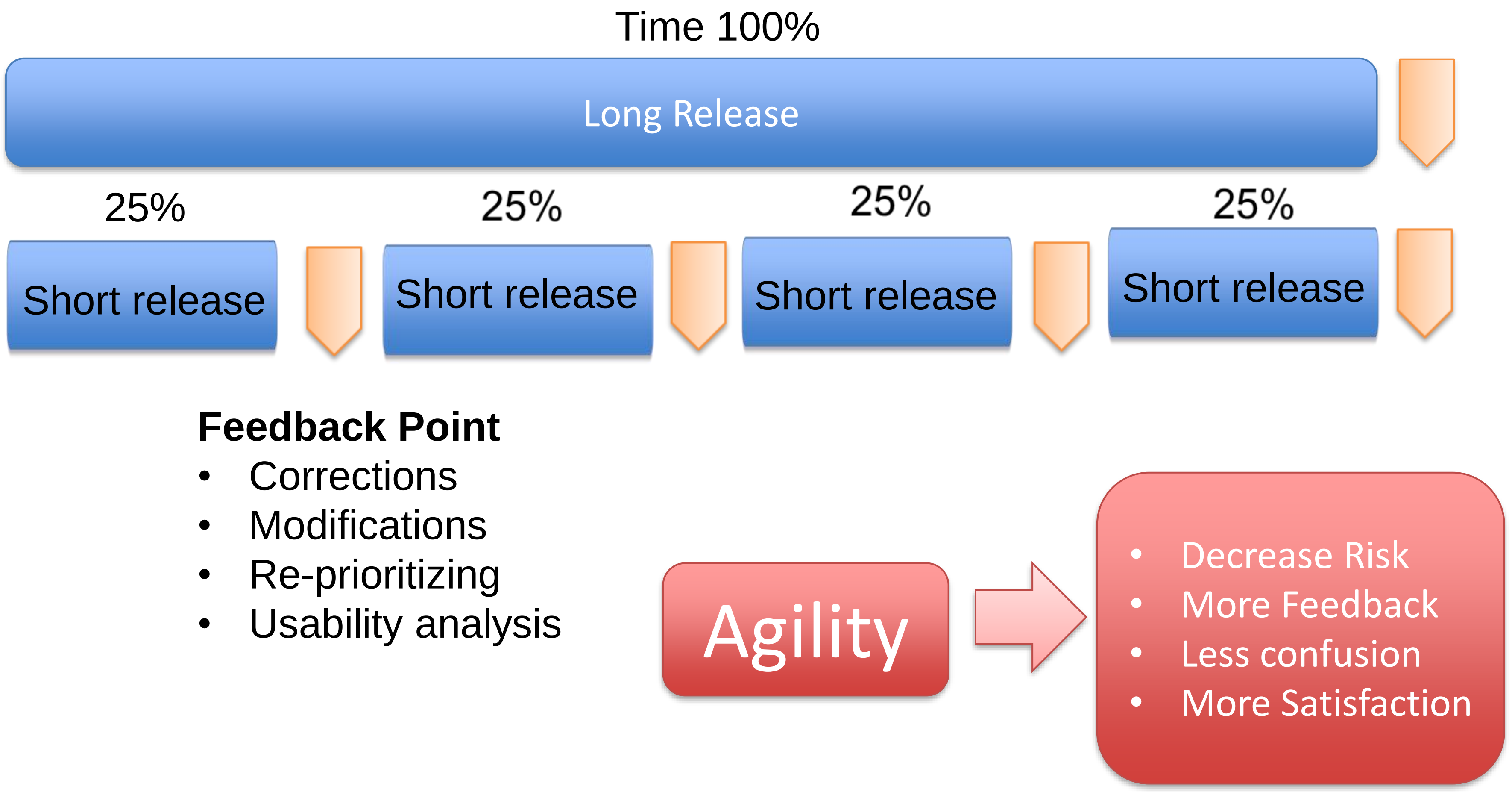
12 Agile Principles

1.	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	7.	Working software is the primary measure of progress.
2.	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	8.	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
3.	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	9.	Continuous attention to technical excellence and good design enhances agility.
4.	Business people and developers must work together daily throughout the project.	10.	Simplicity--the art of maximizing the amount of work not done--is essential.
5.	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.		**12. Team reflection.**
6.	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	12.	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

What is Agile Software Development?



Is Agile better?



Feedback Point

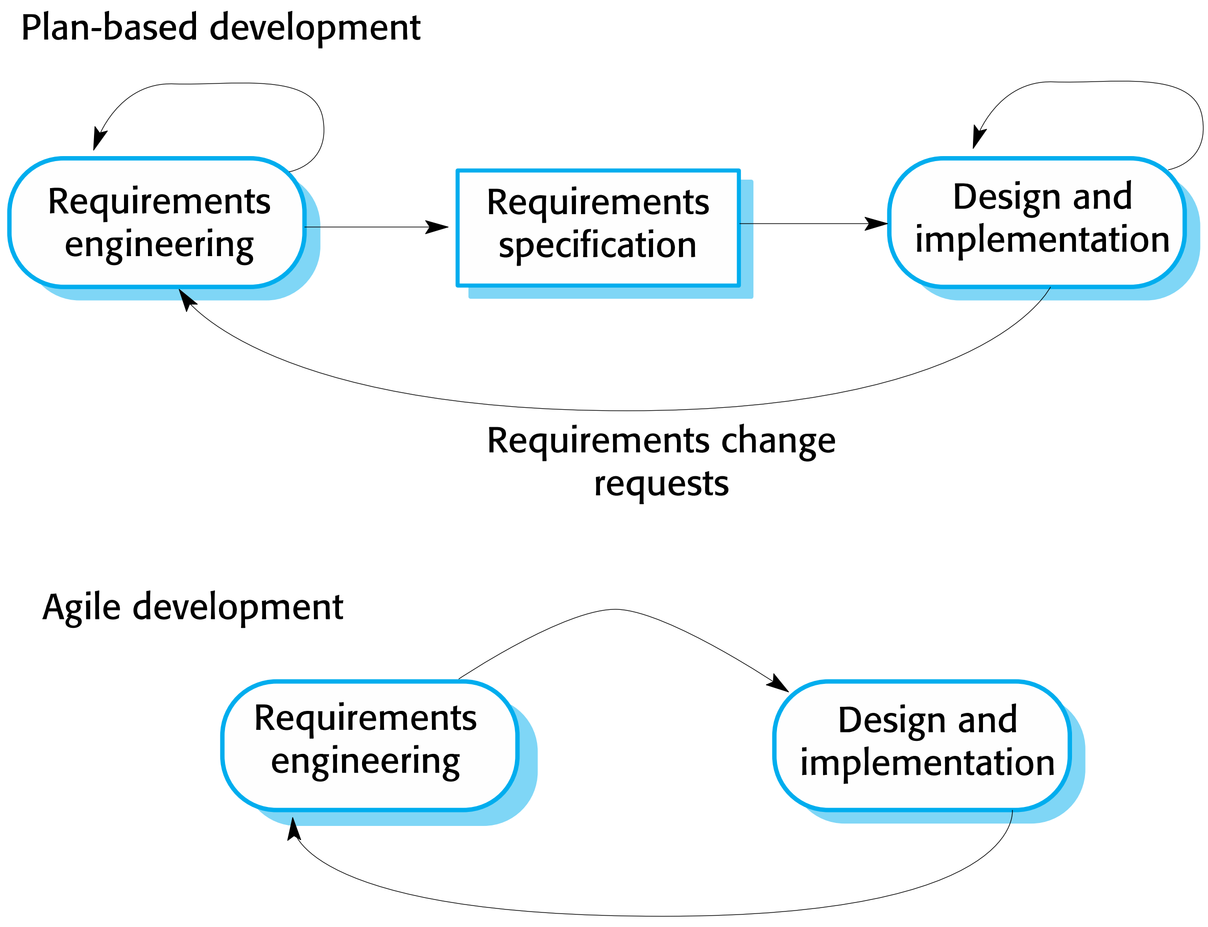
- Corrections
- Modifications
- Re-prioritizing
- Usability analysis

- Decrease Risk
- More Feedback
- Less confusion
- More Satisfaction

What is Agile software development cont.…

- ✧ Program specification, design and implementation are inter-leaved
- ✧ The system is developed as a series of versions or increments with stakeholders involved in version specification and evaluation
- ✧ Frequent delivery of new versions for evaluation
- ✧ Extensive tool support (e.g. automated testing tools) used to support development.
- ✧ Minimal documentation – focus on working code

Plan-driven and agile development


Plan-based development
Requirements engineering
Requirements specification
Design and implementation
Requirements change requests
Agile development
Requirements engineering
Design and implementation

Plan-driven and Agile development

✧ Plan-driven development

- A plan-driven approach to software engineering is based around separate development stages with the outputs to be produced at each of these stages planned in advance.
- Not necessarily waterfall model – plan-driven, incremental development is possible.
- Iteration occurs within activities.

✧ Agile development

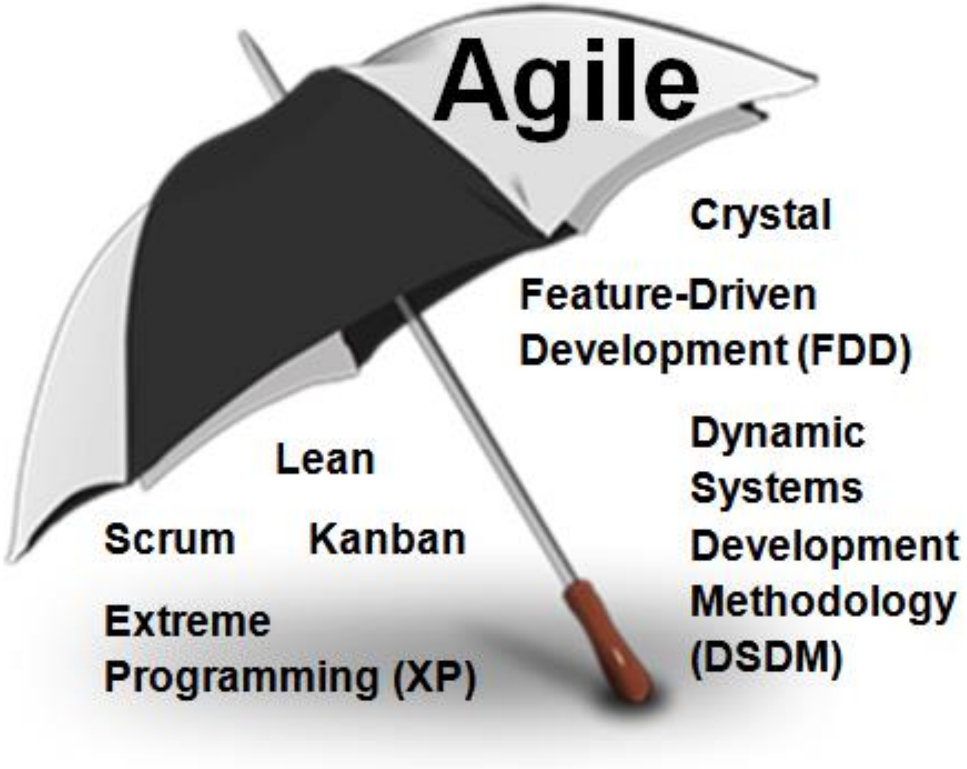
- Specification, design, implementation and testing are inter-leaved and the outputs from the development process are decided through a process of negotiation during the software development process.

What is Agile Software Development?

✧ A mindset

What is Agile Software Development?

✧ An umbrella term for a set of methods and practices based on the values and principles expressed in the Agile Manifesto.


Agile
Crystal
Feature-Driven Development (FDD)
Dynamic Systems Development Methodology (DSDM)
Lean
Scrum
Kanban
Extreme Programming (XP)

Agile Software Development - Successes

- ✧ Small or medium sized product development.
- ✧ Custom system development within organization
- ✧ Experiments in using agile approaches for critical systems engineering.
  - Security, safety, and dependability analysis.
  - Need significant modifications before applying.

Drawbacks / Difficulties

- ✧ Custom representatives are subject to other pressures and cannot take full part in the software development.
- ✧ Individual team members may not have suitable personalities for the intense involvement / team work.
- ✧ Prioritizing changes can be extremely difficult when there are lot of stakeholders.
- ✧ Under pressure from delivery schedules, the team may not have time to carry system simplifications.
- ✧ Cultural change – team members may find it hard to change the practice (informal / defined by team itself)

Plan driven vs Agile

- ✧ Detailed design and specification before moving to implementation? Plan-driven approach
- ✧ Incremental delivery a realistic? Agile
- ✧ The system being development is large? Plan-driven approach
- ✧ Real-time system with complex timing requirements? Plan-driven approach
- ✧ Expected system lifetime?
- ✧ Technology?
- ✧ Development team organization?

Plan driven vs Agile

- ✧ Cultural issues that may affect the system development?
- ✧ Skills of designers and developers?
- ✧ Subjected to any external regulations?

An Executable Software that meets the need and does useful things for the individual user or the organization.