



UNIVERSITY OF SCIENCE
HO CHI MINH CITY

Agile Methods

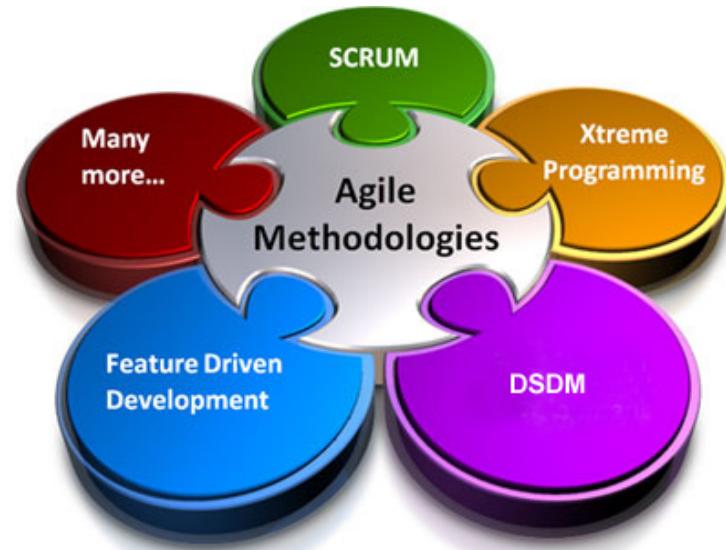
Những pp phát triển pm linh hoạt

Nguyen V. Vu

with some materials adapted from Boehm 2003

Outline

- Agile Manifesto
- Extreme Programming
- Scrum
- Summary



The Agile Manifesto

tuyên ngôn

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.

The Agile Manifesto (cont'd)

- Our 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

The Agile Manifesto (cont'd)

- 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**
- **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

The Agile Manifesto (cont'd)

- 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**
nhóm tương tác tốt => tạo ra những phần mềm tốt ...
- At regular intervals, the team **reflects** on how to become more effective, then tunes and **adjusts** its behavior accordingly

Outline

- Introduction to Agile Methods
- Agile Manifesto
- Extreme Programming Method
 - Principles
 - The 12 practices
- Scrum Method
- Summary

XP Principles

- Philosophy: *Take known good practices and push them to extremes*
- “If code reviews are good, we’ll review code all the time”
- “If testing is good, we’ll test all the time”
- “If design is good, we’ll make it part of everybody’s daily business”

XP Principles (cont'd)

- “If simplicity is good, we'll always leave the system with the simplest design that supports its current functionality”
- “If architecture is important, everybody will work defining and refining the architecture all the time”
- “If integration testing is important, then we'll integrate and test several times a day”

XP Principles (cont'd)

- “If short iterations are good, we’ll make the iterations really, really short – seconds and minutes and hours, not weeks and months and years”
- “If customer involvement is good, we’ll make them full-time participants”

XP: The 12 Practices

- The Planning Game
- Small Releases
- Metaphor
- Simple Design
- Testing
- Refactoring
- Pair Programming
- Collective Ownership
- Continuous Integration
- 40-hour Week
- On-site Customer
- Coding Standards

☞ **Use these practices generatively, not imperatively**

The Planning Game

planning game là gì, thực hành ra sao

- Use stories to facilitate knowledge transfer
- Put decisions in the hands of the person with the best knowledge:
 - business decisions → Customer
 - software decisions → Developer
- Plan only as far as your knowledge allows
 - next iteration or next release



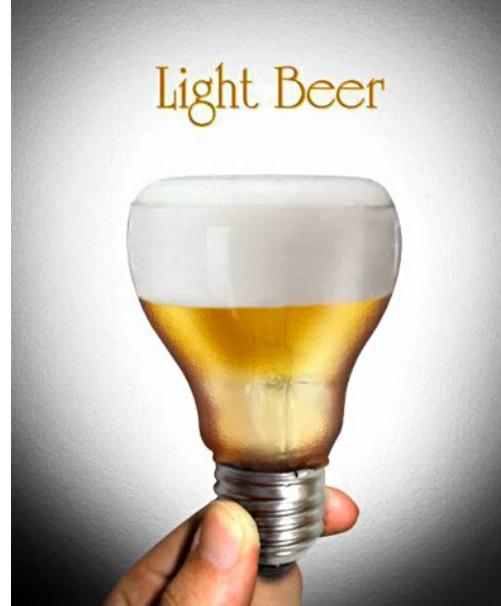
Small Releases

- Supports quick feedback from users
- Simplify the tracking of metrics
 - stories per iteration → project velocity
- Increase the manageability of the project for the customer
 - But complicate user conservation of familiarity

Metaphor

mượn những thứ trong thực tế vào hệ thống máy tính vd như: print, cut, cửa sổ, thùng rác ..

- Ground all discussions in a single shared story of how the whole system works
- Provide an overarching view of the project
- Connect program to work process



Simple Design

vì sao cần phải thực hiện simple design, ý nghĩa của việc sd

- Design embodies only the needed complexity and no more
 - emphasis on top-down or bottom-up design as needed to meet this iteration's stories
 - extra complexity removed when discovered
- Simpler designs are easier to modify, maintain, and describe
 - decreases the cost of design changes
 - but no notion of product line architecture

Testing

- Unit tests verify the programmer's work
 - must be done by programmer
 - constant testing makes finding new bugs faster and easier
- Functional tests verify that a story is complete
 - developed by customer
 - tests define functional requirements

Refactoring

refactoring là gì? vì sao phải refactoring

refactoring thường xuyên hay lâu lâu làm f1 lần để tốt hơn

- Procedure for implementing iterative design
 - ❑ behavior-preserving
 - ❑ improves communication among developers
 - ❑ adds flexibility to the programming process
- Design is important – do it all the time
 - ❑ software development process is a design process
 - ❑ But redesign much more expensive for large systems

Pair Programming

pp là gì? lợi ích của pair programming

- All code is written by two programmers at a single machine
- Inspections are important, so do them all the time
- Increase implicit knowledge transfer
- Decrease cycle time, but increase effort



Collective Ownership

làm chủ tập thể - mang lại lợi ích gì

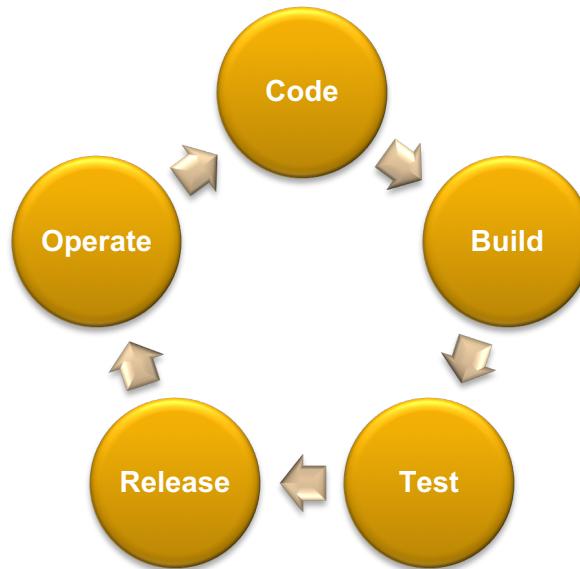
- Everyone owns all of the code
 - anyone can change any code anywhere
 - no personal ownership of modules
 - no egoless programming either
- Everyone is permitted access to all the code so everyone has a stake in knowing all of the code (that they will work with)

Continuous Integration

tích hợp liên tục - là gì

tích hợp sản phẩm liên tục, nhiều người code => đưa lên hệ thống => build => kiểm thử => released
giảm thiểu những

- The system always works
 - there is always something to be released
- Similar to rapid releases
 - fast feedback to developers on problems
 - no ‘big bang’ integration disasters



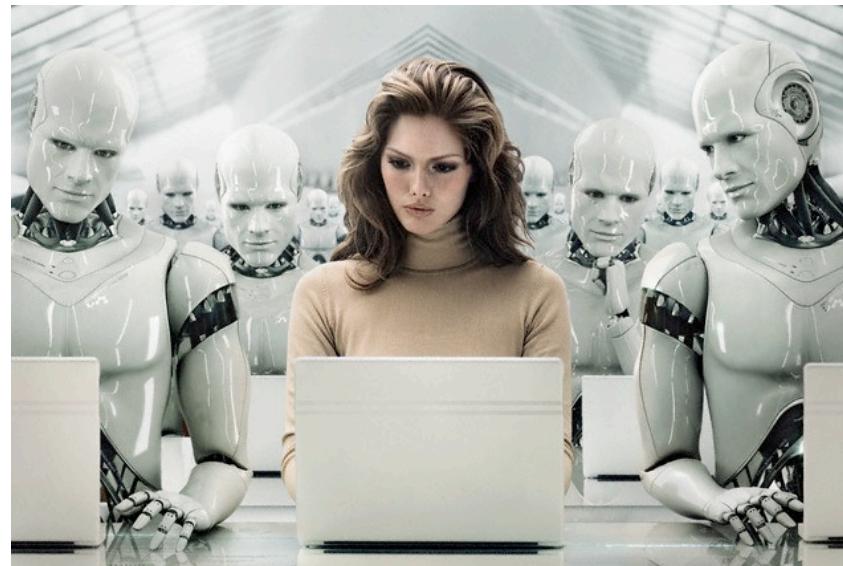
40-hour Week

- No heroes
- Knowledge can only be transferred at a limited rate
- Work for sustained speed, not a single sprint
 - never work overtime a second week in a row



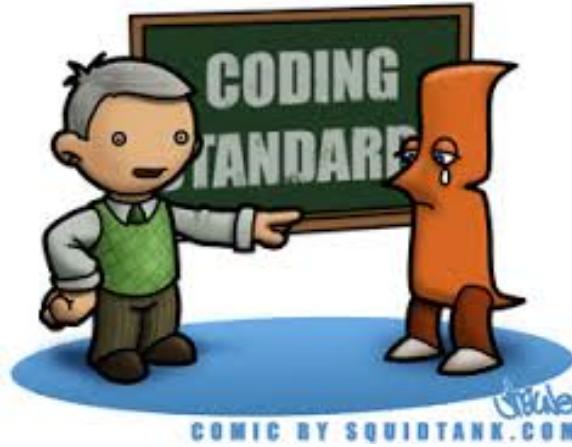
On-site Customer

- A real, live user available full-time to answer questions as they occur
- Programmers don't know everything
- Business knowledge is the key to a successful business project



Coding Standards

- Communication occurs through the code
- Common standard promotes understanding of other developers' code
- Helps promote team focus



Outline

- Agile Manifesto
- Extreme Programming Method
- Scrum
 - Concepts
 - Practices
- Summary

vì sao agile lại phù hợp với những dự án nhỏ?

safety-critical system

- hệ thống xe tự hành: lỗi thì hệ thống gây chết người
- hệ thống y khoa: điều trị ung thư (bắn tia laser..) cần đảm bảo an toàn cao

What is Scrum?

- An agile method that employs a set of simple practices and rules to incrementally develop products
- Developed by Ken Schwaber and Jeff Sutherland
 - vòng lặp (sprint) cố định



This Photo by Unknown Author is licensed under [CC BY](#)



Scrum Concepts

■ Daily Scrum **hợp giao ban hàng ngày**

- a short daily meeting (less than 30 minutes) for the team to monitor status and communicate problems

■ Sprint **tương tự như iteration (từ lúc bắt đầu đến lúc kết thúc giai đoạn), thời gian sprint là cố định (time boxed iteration)**

- a development cycle (typically 30 days)

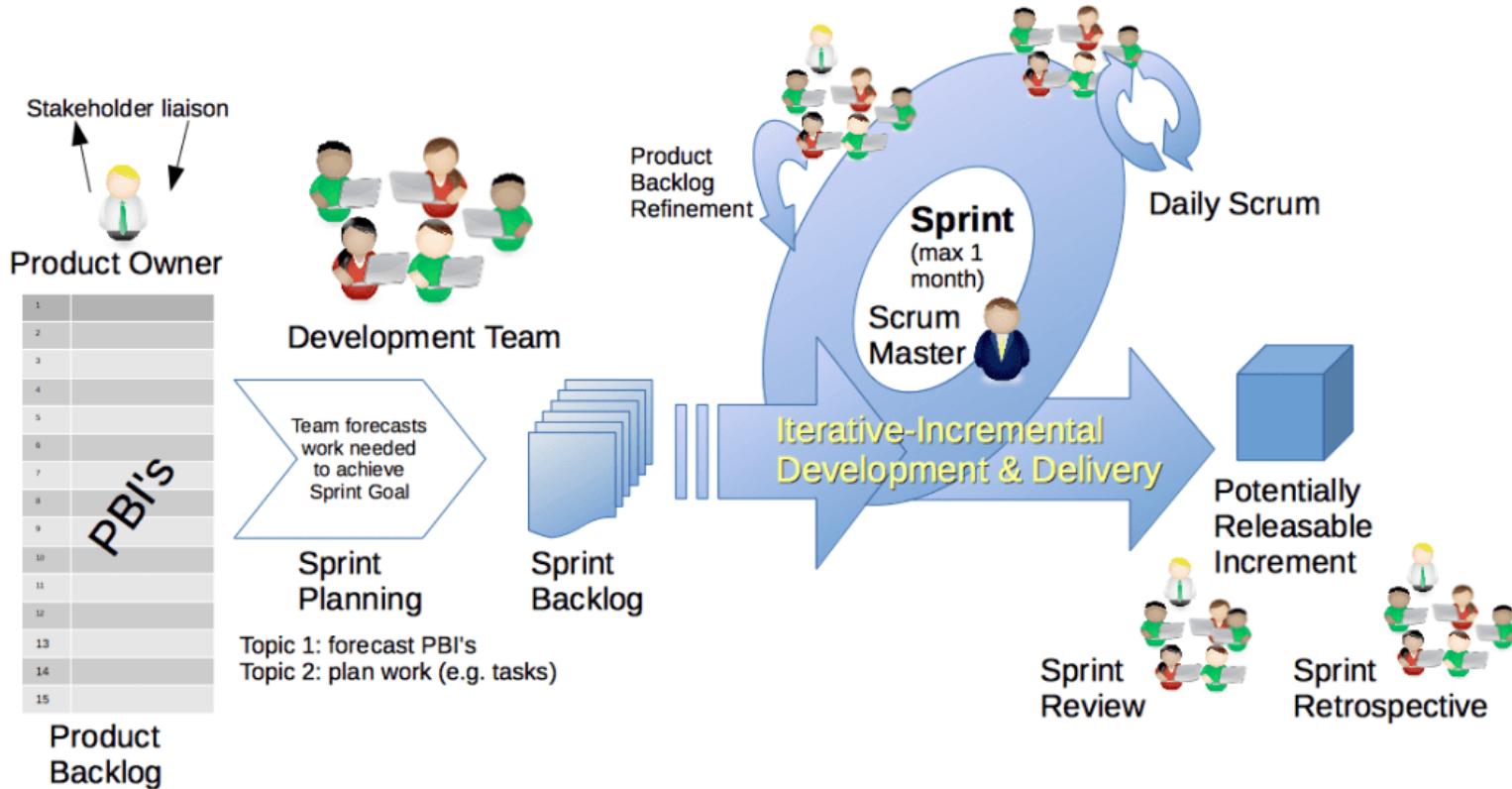
■ Backlog **khối cv cần phải làm**

- Product backlog: prioritized list of product requirements **những yc phần mềm mình cần phải làm**
- Sprint backlog: prioritized list of requirements allocated to the sprint **những yc phải làm cho mỗi sprint/work items/ cản trở**
- Impediments backlog: list of issues

■ Burndown chart **biểu đồ diễn đạt tiến độ đồ án**

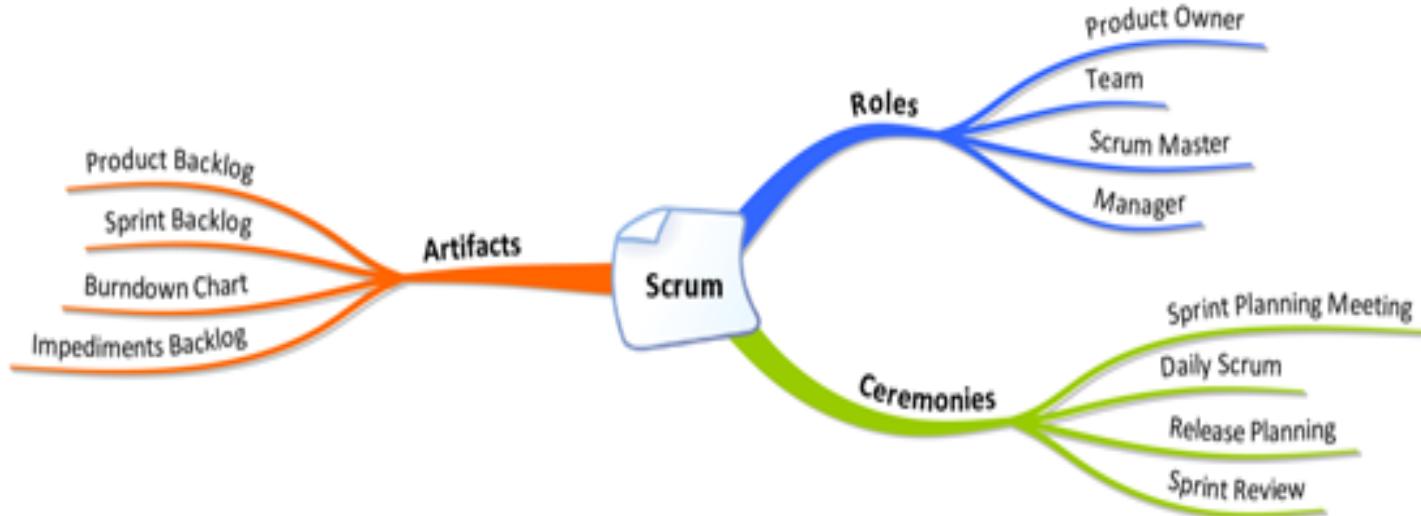
- chart showing the progress (backlog items completed)

Scrum at a Glance



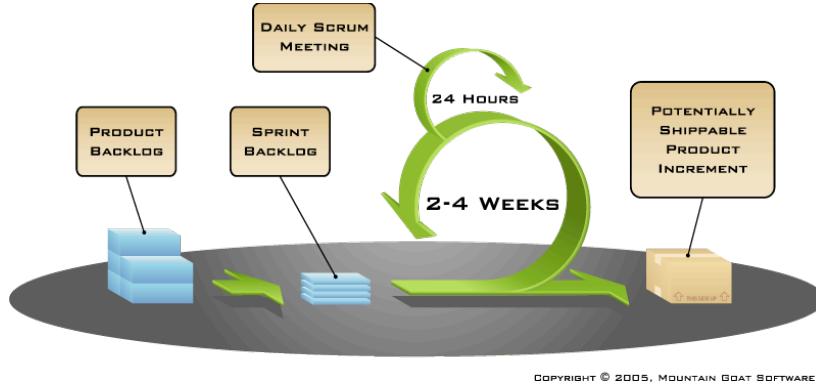
[This Photo](#) by Unknown Author is licensed under [CC BY-NC-ND](#)

Scrum Process (cont'd)



Artifacts

# in PM-Pres.	Official P/B-Area Description	PM Rank	IT Rank	Total Rank	Business Value
5.1	PROCESS E	4	4	1	25
2.4	PROCESS B	6	1	1	25
2.1	PROCESS X	7	1	3	15
IIQ0074	PROCESS D	2		2	15
PI0060	PROCESS M	2		2	2
2.2	PROCESS N	8	1	3	25
3.1/3.3	PROCESS S	1	1	5	80
3.6	PROCESS A	2	2	6	60
6.2	PROCESS Y	1	1	4	15
3.1/3.3	Subprocess 5.1			5	50
3.1/3.3	Subprocess 5.2			5	50
3.1/3.3	Subprocess 5.3			5	35
3.1/3.3	Subprocess 5.4			5	51
3.1/3.3	Subprocess 5.5			5	45



Product Backlog:

- Overall product requirements
- Items are estimated and prioritized

Project: Demo - Server (fhem) - Query: Current Sprint Backlog Items									
#	Title	Description	Work Item	Status	Task Points	Assigned To	Due Date	Completed	Created Date
1	1288 10-Bug 1.2	1	6 Done	1	1	bug 1.2	2023-08-15	1	2023-08-15 10:00:00
2	1288 10-Bug 1.1	2	6 Done	1	1	colin.ledard	bug 1.1 sprint task 1	1	2023-08-15 10:00:00
3	1344 Fix issue - Related to work item 332	4	6 Done	1	1	parker.campbell	updated from my task board again	1	2023-08-15 10:00:00
4	1355 Fix Stressed Proc - Related to work item	1	6 Done	1	1	david.wright	1	1	2023-08-15 10:00:00
5	1363 10-Bug 1.2	In Progress	1	1	1	alexander.schmid	1. test caption sprint backlog item, fix 1	1	2023-08-15 10:00:00
6	1288 10-Bug 1.2	5	6 Done	2	2	bug 1.2 sprint task 2	2023-08-15	1	2023-08-15 10:00:00
7	1288 10-Bug 1.2	6	6 Done	2	2	bug 1.2 sprint task 2	2023-08-15	1	2023-08-15 10:00:00
8	1288 10-Bug 1.2	7	6 Done	3	3	bug 1.2 sprint task 3	2023-08-15	1	2023-08-15 10:00:00
9	1288 10-Bug 1.1	8	6 Done	3	3	bug 1.1 sprint task 1	2023-08-15	1	2023-08-15 10:00:00
10	1288 10-Bug 1.1	9	6 Done	4	4	bug 1.1 sprint task 2	2023-08-15	1	2023-08-15 10:00:00
11	1288 10-Bug 1.1	10	6 Done	4	4	bug 1.1 sprint task 4	2023-08-15	1	2023-08-15 10:00:00
12	1288 10-Bug 1.1	11	6 Done	4	4	bug 1.1 sprint task 6	2023-08-15	1	2023-08-15 10:00:00
13	1288 10-Bug 1.1	12	6 Done	4	4	bug 1.1 sprint task 8	2023-08-15	1	2023-08-15 10:00:00
14	1288 10-Bug 1.1	13	6 Done	4	4	bug 1.1 sprint task 10	2023-08-15	1	2023-08-15 10:00:00
15	1288 10-Bug 1.1	14	6 Done	4	4	bug 1.1 sprint task 12	2023-08-15	1	2023-08-15 10:00:00
16	1288 10-Bug 1.1	15	6 Done	4	4	bug 1.1 sprint task 14	2023-08-15	1	2023-08-15 10:00:00
17	1288 10-Bug 1.1	16	6 Done	4	4	bug 1.1 sprint task 16	2023-08-15	1	2023-08-15 10:00:00
18	1288 10-Bug 1.1	17	6 Done	4	4	bug 1.1 sprint task 18	2023-08-15	1	2023-08-15 10:00:00
19	1288 10-Bug 1.1	18	6 Done	4	4	bug 1.1 sprint task 20	2023-08-15	1	2023-08-15 10:00:00
20	1288 10-Bug 1.1	19	6 Done	4	4	bug 1.1 sprint task 22	2023-08-15	1	2023-08-15 10:00:00
21	1288 10-Bug 1.1	20	6 Done	4	4	bug 1.1 sprint task 24	2023-08-15	1	2023-08-15 10:00:00
22	1288 10-Bug 1.1	21	6 Done	4	4	bug 1.1 sprint task 26	2023-08-15	1	2023-08-15 10:00:00
23	1288 10-Bug 1.1	22	6 Done	4	4	bug 1.1 sprint task 28	2023-08-15	1	2023-08-15 10:00:00
24	1288 10-Bug 1.1	23	6 Done	4	4	bug 1.1 sprint task 30	2023-08-15	1	2023-08-15 10:00:00
25	1288 10-Bug 1.1	24	6 Done	4	4	bug 1.1 sprint task 32	2023-08-15	1	2023-08-15 10:00:00
26	1288 10-Bug 1.1	25	6 Done	4	4	bug 1.1 sprint task 34	2023-08-15	1	2023-08-15 10:00:00
27	1288 10-Bug 1.1	26	6 Done	4	4	bug 1.1 sprint task 36	2023-08-15	1	2023-08-15 10:00:00
28	1288 10-Bug 1.1	27	6 Done	4	4	bug 1.1 sprint task 38	2023-08-15	1	2023-08-15 10:00:00
29	1288 10-Bug 1.1	28	6 Done	4	4	bug 1.1 sprint task 40	2023-08-15	1	2023-08-15 10:00:00
30	1288 10-Bug 1.1	29	6 Done	4	4	bug 1.1 sprint task 42	2023-08-15	1	2023-08-15 10:00:00
31	1288 10-Bug 1.1	30	6 Done	4	4	bug 1.1 sprint task 44	2023-08-15	1	2023-08-15 10:00:00
32	1288 10-Bug 1.1	31	6 Done	4	4	bug 1.1 sprint task 46	2023-08-15	1	2023-08-15 10:00:00
33	1288 10-Bug 1.1	32	6 Done	4	4	bug 1.1 sprint task 48	2023-08-15	1	2023-08-15 10:00:00
34	1288 10-Bug 1.1	33	6 Done	4	4	bug 1.1 sprint task 50	2023-08-15	1	2023-08-15 10:00:00
35	1288 10-Bug 1.1	34	6 Done	4	4	bug 1.1 sprint task 52	2023-08-15	1	2023-08-15 10:00:00
36	1288 10-Bug 1.1	35	6 Done	4	4	bug 1.1 sprint task 54	2023-08-15	1	2023-08-15 10:00:00
37	1288 10-Bug 1.1	36	6 Done	4	4	bug 1.1 sprint task 56	2023-08-15	1	2023-08-15 10:00:00
38	1288 10-Bug 1.1	37	6 Done	4	4	bug 1.1 sprint task 58	2023-08-15	1	2023-08-15 10:00:00
39	1288 10-Bug 1.1	38	6 Done	4	4	bug 1.1 sprint task 60	2023-08-15	1	2023-08-15 10:00:00
40	1288 10-Bug 1.1	39	6 Done	4	4	bug 1.1 sprint task 62	2023-08-15	1	2023-08-15 10:00:00
41	1288 10-Bug 1.1	40	6 Done	4	4	bug 1.1 sprint task 64	2023-08-15	1	2023-08-15 10:00:00
42	1288 10-Bug 1.1	41	6 Done	4	4	bug 1.1 sprint task 66	2023-08-15	1	2023-08-15 10:00:00
43	1288 10-Bug 1.1	42	6 Done	4	4	bug 1.1 sprint task 68	2023-08-15	1	2023-08-15 10:00:00
44	1288 10-Bug 1.1	43	6 Done	4	4	bug 1.1 sprint task 70	2023-08-15	1	2023-08-15 10:00:00
45	1288 10-Bug 1.1	44	6 Done	4	4	bug 1.1 sprint task 72	2023-08-15	1	2023-08-15 10:00:00
46	1288 10-Bug 1.1	45	6 Done	4	4	bug 1.1 sprint task 74	2023-08-15	1	2023-08-15 10:00:00
47	1288 10-Bug 1.1	46	6 Done	4	4	bug 1.1 sprint task 76	2023-08-15	1	2023-08-15 10:00:00
48	1288 10-Bug 1.1	47	6 Done	4	4	bug 1.1 sprint task 78	2023-08-15	1	2023-08-15 10:00:00
49	1288 10-Bug 1.1	48	6 Done	4	4	bug 1.1 sprint task 80	2023-08-15	1	2023-08-15 10:00:00
50	1288 10-Bug 1.1	49	6 Done	4	4	bug 1.1 sprint task 82	2023-08-15	1	2023-08-15 10:00:00
51	1288 10-Bug 1.1	50	6 Done	4	4	bug 1.1 sprint task 84	2023-08-15	1	2023-08-15 10:00:00
52	1288 10-Bug 1.1	51	6 Done	4	4	bug 1.1 sprint task 86	2023-08-15	1	2023-08-15 10:00:00
53	1288 10-Bug 1.1	52	6 Done	4	4	bug 1.1 sprint task 88	2023-08-15	1	2023-08-15 10:00:00
54	1288 10-Bug 1.1	53	6 Done	4	4	bug 1.1 sprint task 90	2023-08-15	1	2023-08-15 10:00:00
55	1288 10-Bug 1.1	54	6 Done	4	4	bug 1.1 sprint task 92	2023-08-15	1	2023-08-15 10:00:00
56	1288 10-Bug 1.1	55	6 Done	4	4	bug 1.1 sprint task 94	2023-08-15	1	2023-08-15 10:00:00
57	1288 10-Bug 1.1	56	6 Done	4	4	bug 1.1 sprint task 96	2023-08-15	1	2023-08-15 10:00:00
58	1288 10-Bug 1.1	57	6 Done	4	4	bug 1.1 sprint task 98	2023-08-15	1	2023-08-15 10:00:00
59	1288 10-Bug 1.1	58	6 Done	4	4	bug 1.1 sprint task 100	2023-08-15	1	2023-08-15 10:00:00
60	1288 10-Bug 1.1	59	6 Done	4	4	bug 1.1 sprint task 102	2023-08-15	1	2023-08-15 10:00:00
61	1288 10-Bug 1.1	60	6 Done	4	4	bug 1.1 sprint task 104	2023-08-15	1	2023-08-15 10:00:00
62	1288 10-Bug 1.1	61	6 Done	4	4	bug 1.1 sprint task 106	2023-08-15	1	2023-08-15 10:00:00
63	1288 10-Bug 1.1	62	6 Done	4	4	bug 1.1 sprint task 108	2023-08-15	1	2023-08-15 10:00:00
64	1288 10-Bug 1.1	63	6 Done	4	4	bug 1.1 sprint task 110	2023-08-15	1	2023-08-15 10:00:00
65	1288 10-Bug 1.1	64	6 Done	4	4	bug 1.1 sprint task 112	2023-08-15	1	2023-08-15 10:00:00
66	1288 10-Bug 1.1	65	6 Done	4	4	bug 1.1 sprint task 114	2023-08-15	1	2023-08-15 10:00:00
67	1288 10-Bug 1.1	66	6 Done	4	4	bug 1.1 sprint task 116	2023-08-15	1	2023-08-15 10:00:00
68	1288 10-Bug 1.1	67	6 Done	4	4	bug 1.1 sprint task 118	2023-08-15	1	2023-08-15 10:00:00
69	1288 10-Bug 1.1	68	6 Done	4	4	bug 1.1 sprint task 120	2023-08-15	1	2023-08-15 10:00:00
70	1288 10-Bug 1.1	69	6 Done	4	4	bug 1.1 sprint task 122	2023-08-15	1	2023-08-15 10:00:00
71	1288 10-Bug 1.1	70	6 Done	4	4	bug 1.1 sprint task 124	2023-08-15	1	2023-08-15 10:00:00
72	1288 10-Bug 1.1	71	6 Done	4	4	bug 1.1 sprint task 126	2023-08-15	1	2023-08-15 10:00:00
73	1288 10-Bug 1.1	72	6 Done	4	4	bug 1.1 sprint task 128	2023-08-15	1	2023-08-15 10:00:00
74	1288 10-Bug 1.1	73	6 Done	4	4	bug 1.1 sprint task 130	2023-08-15	1	2023-08-15 10:00:00
75	1288 10-Bug 1.1	74	6 Done	4	4	bug 1.1 sprint task 132	2023-08-15	1	2023-08-15 10:00:00
76	1288 10-Bug 1.1	75	6 Done	4	4	bug 1.1 sprint task 134	2023-08-15	1	2023-08-15 10:00:00
77	1288 10-Bug 1.1	76	6 Done	4	4	bug 1.1 sprint task 136	2023-08-15	1	2023-08-15 10:00:00
78	1288 10-Bug 1.1	77	6 Done	4	4	bug 1.1 sprint task 138	2023-08-15	1	2023-08-15 10:00:00
79	1288 10-Bug 1.1	78	6 Done	4	4	bug 1.1 sprint task 140	2023-08-15	1	2023-08-15 10:00:00
80	1288 10-Bug 1.1	79	6 Done	4	4	bug 1.1 sprint task 142	2023-08-15	1	2023-08-15 10:00:00
81	1288 10-Bug 1.1	80	6 Done	4	4	bug 1.1 sprint task 144	2023-08-15	1	2023-08-15 10:00:00
82	1288 10-Bug 1.1	81	6 Done	4	4	bug 1.1 sprint task 146	2023-08-15	1	2023-08-15 10:00:00
83	1288 10-Bug 1.1	82	6 Done	4	4	bug 1.1 sprint task 148	2023-08-15	1	2023-08-15 10:00:00
84	1288 10-Bug 1.1	83	6 Done	4	4	bug 1.1 sprint task 150	2023-08-15	1	2023-08-15 10:00:00
85	1288 10-Bug 1.1	84	6 Done	4	4	bug 1.1 sprint task 152	2023-08-15	1	2023-08-15 10:00:00
86	1288 10-Bug 1.1	85	6 Done	4	4	bug 1.1 sprint task 154	2023-08-15	1	2023-08-15 10:00:00
87	1288 10-Bug 1.1	86	6 Done	4	4	bug 1.1 sprint task 156	2023-08-15	1	2023-08-15 10:00:00
88	1288 10-Bug 1.1	87	6 Done	4	4	bug 1.1 sprint task 158	2023-08-15	1	2023-08-15 10:00:00
89	1288 10-Bug 1.1	88	6 Done	4	4	bug 1.1 sprint task 160	2023-08-15	1	2023-08-15 10:00:00
90	1288 10-Bug 1.1	89	6 Done	4	4	bug 1.1 sprint task 162	2023-08-15	1	2023-08-15 10:00:00
91	1288 10-Bug 1.1	90	6 Done	4	4	bug 1.1 sprint task 164	2023-08-15	1	2023-08-15 10:00:00
92	1288 10-Bug 1.1</								

Scrum Roles



- Facilitate Scrum practices
- Enforce Scrum principles



Product Owner
5/19/22

- Create vision and requirements
- Contact point
- Create releases plan



Team

- Estimate Product backlog
- Create Sprint backlog
- Implement backlog items



Manager

- Ensure resources available
- Resources management
- Team building

Activities

- Release planning
 - Prioritize and allocate features to releases
- Sprint planning
 - Set goal and allocate backlog items to Sprint
- Daily Scrum
 - Review daily status
- Sprint review
 - Held at end of Sprint
 - Review features completed
 - Decide what went wrong, right



Outline

- Introduction to Agile Methods
- Agile Manifesto
- Extreme Programming Method
- Scrum Method
- Summary
 - Strengths and Weaknesses



Where Agile Methods Work Best?

- Small teams (2 – 20 people)
- Short iterations (1 – 2 weeks)
- Collocated teams – not recommended for distributed teams
- Non safety-critical systems

Strengths and Weaknesses of Agile Methods

■ Strengths

- Suitable for dynamic, chaotic environments
- Supportive of emergent requirements, rapid change
- Avoiding documentation overhead
- Empowering/encouraging people

■ Weaknesses

- Requiring high-capable, self-motivated people
- Difficult in safety-critical systems
- Problem in large projects