W4156

Methodology

Agenda

- Defining Methodology
- History Lesson: The ‘Crisis’
- Agile
- Comparison

Chaos?

What if we started a new project tomorrow with 5 engineers.

- What would they do day 1, 2, 10, 30?
- If/when would they do any activities (requirements, design, test, code, deliver, receive feedback?)
- When would the software be ready?
- Would everyone do the same job or would they specialize?
- Which tasks would everyone work on?

Methodology

Methodology: Within a project to develop/enhance software the methodology defines

1. **Process:** the process to structure and plan the key activities
2. **People:** interaction, roles and responsibilities of *project participants* (user + engineer)
3. **Practices:** (may also define) technical practices/tooling

(There are many floaty definitions for methodology. Many focus primarily on *process* which for reasons we will see is deficient)

(I am going to tease apart 'methodology' which we will cover today and 'method' which we will cover tomorrow. Crudely, 'methodology' is the set of guiding principles, philosophy/theology about to to produce software. We will then discuss 'methods' which are more practical, executable activies, roles and practices)

Importance of Methodology

From our discussion we can see that a lack of methodology could impact our success.

What are the characteristics of a ‘good’ methodology …

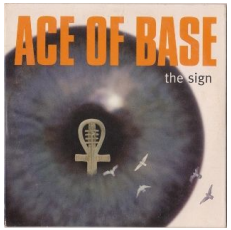
- Decrease time to market/value (quick)
- Promote[1] building the ‘right thing’
- Promote building it the ‘right way’ / good design
- Promote developers spends as much time as possible writing code[2]
- Cost effective
- ….

[1] A methodology is not a guarantee
[2] Time will be spent on non-dev activities. Is this time well spent?

History Lesson

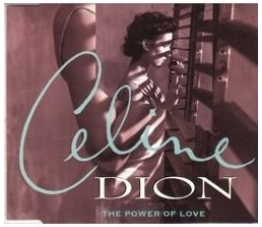
The year was 1994
And the world was in crisis

Billboard top 5 included...



“The Sign”

“Hero”



“The Power of Love”

1994 Billboard

“Rachel finds out” in the season 1 finale ...

Source

And yet, the more important crisis was the *"Software Crisis"*

In 1994 of a wide survey[1] conducted by the Standish group what percentage of projects were:

Percentage of Projects	%
A success (on time and budget with original feature set)	
Challenged (complete but over budget and/or under featured)	
Cancelled during development	

[1] There is a challenge to industrial software engineering research (response bias, lack of repeatability, etc)

Source: Standish Group Chaos Report 1994

However, more important crisis was the *"Software Crisis"*

Percentage of Projects	%
A success (on time and budget with original feature set)	**16**
Challenged (complete but over budget and/or under featured)	**53**
Cancelled during development	**31**

And the US was spending $250bn p/a on software .....

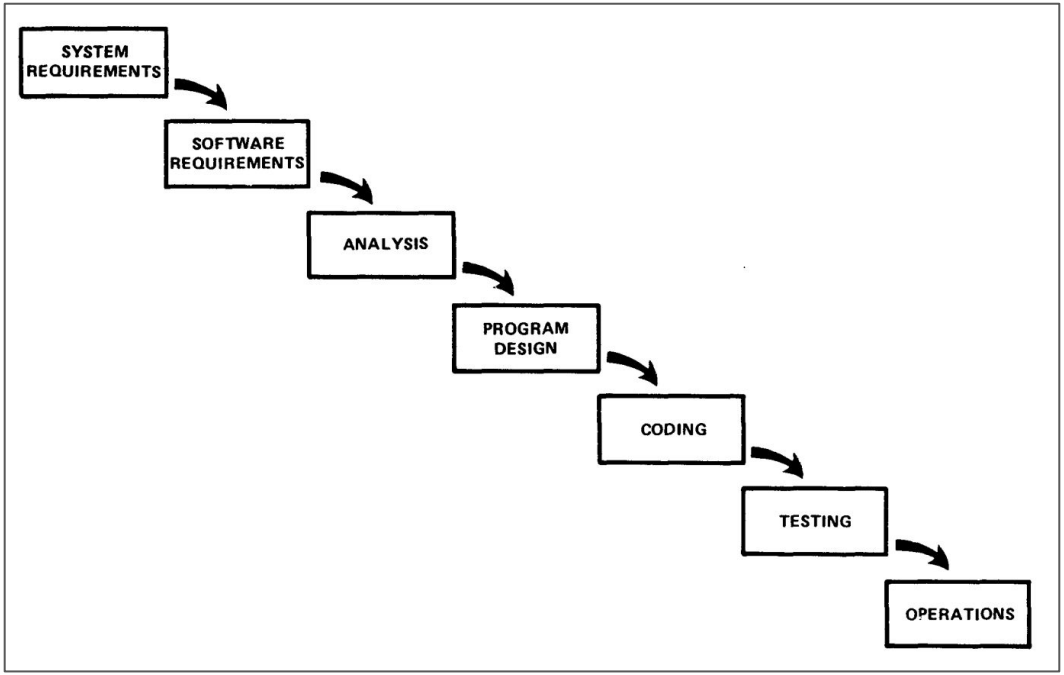
Source: Standish Group Chaos Report 1994

What was going on?

1. Software Engineering is still an incredibly young discipline (so lets be kind to ourselves)
2. There was a relative explosion in computing power
3. There was an attempt to apply thinking from other engineering disciplines to software. Specifically *plan based predictive* approaches

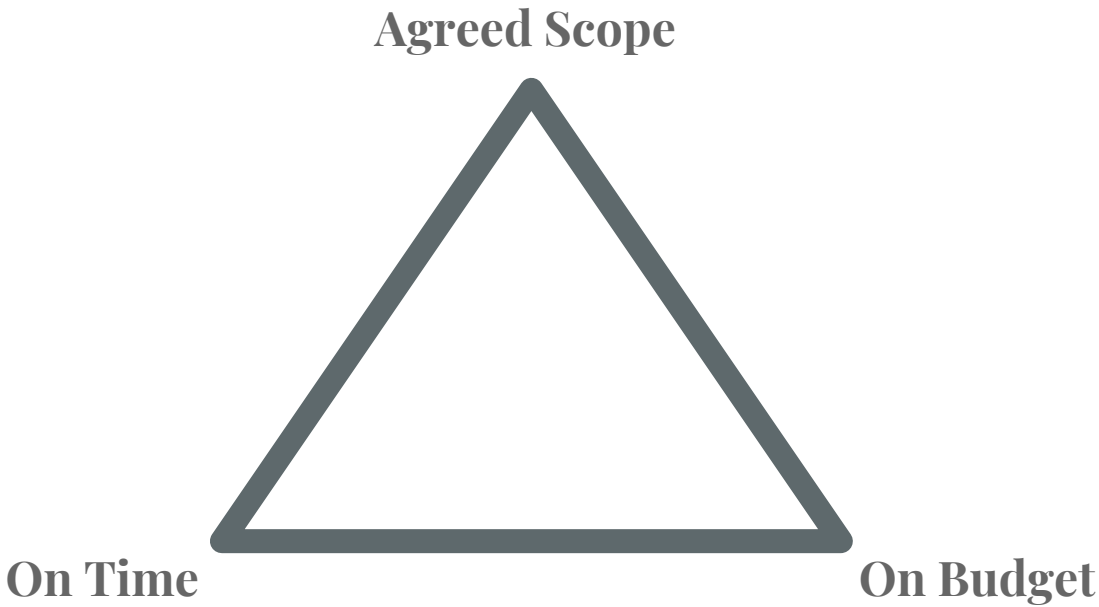
Predictive Approaches

Predictive/Waterfall: Process Perspective



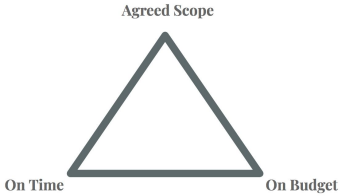
Royce [ICSE '87 Proceedings of the 9th international conference on Software Engineering]. Note - Royce is sometimes incorrectly cited as the 'inventor' of Waterfall. He was actually *critiquing* methods/projects he had observed

Predictive Project Management: Iron Triangle



Iron Triangle

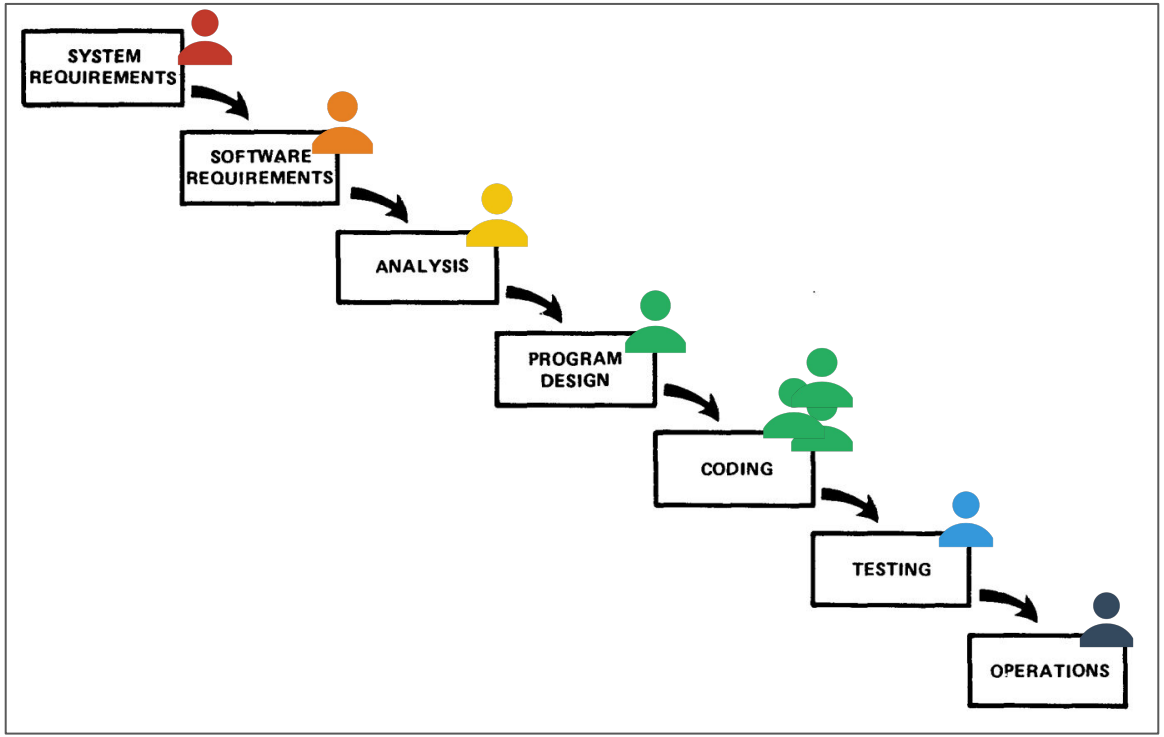
Predictive:
Project Management Thinking



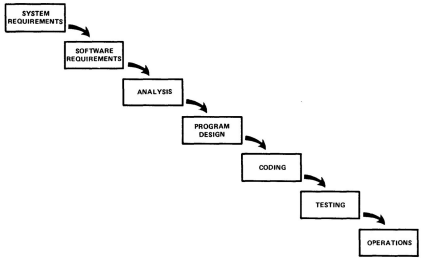
Q1: We deliver what was *originally agreed* but we knew mid-project it provides no value …. Is “on time, scope and budget” a success …..?

Q2: Where does internal quality attributes appear in the triangle?

Predictive: People Perspective



Predictive/Waterfall: Process Discussion



Can you identify any challenges created/amplified by a predictive approach?

Adaptive Approaches

(And how a ski trip coined a movement)

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.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

Agile Manifesto

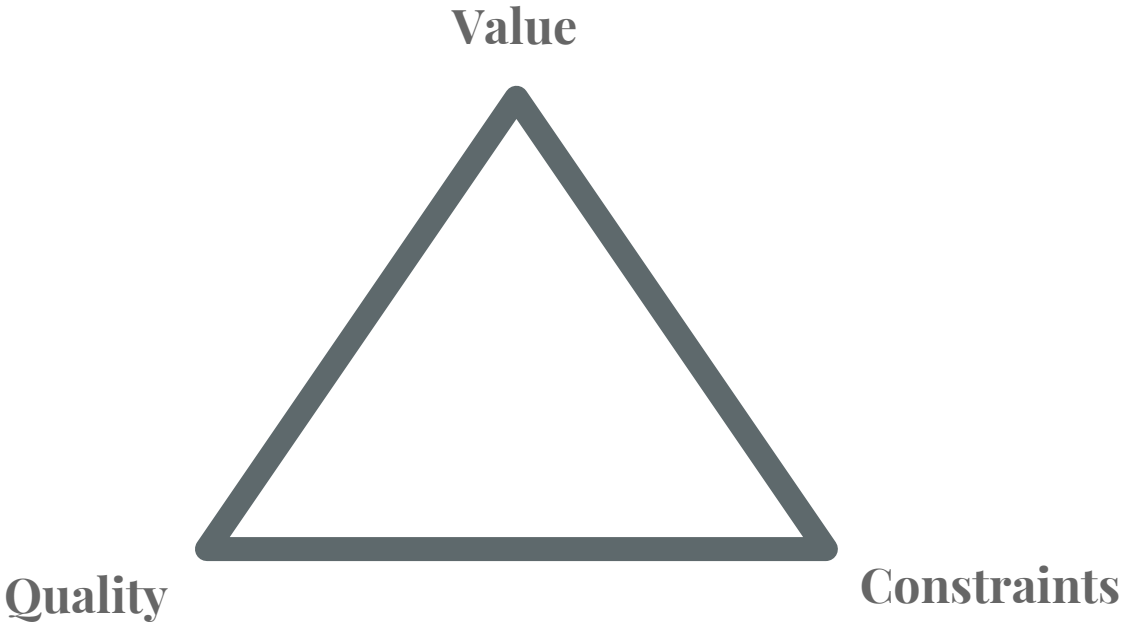
Much confusion/ semantic diffusions on “Agile”

The *manifesto* is not (nor was intended to be) a “concrete” software development process to follow. (even further - a singular ‘checklist’ runs *contrary* to Agile)

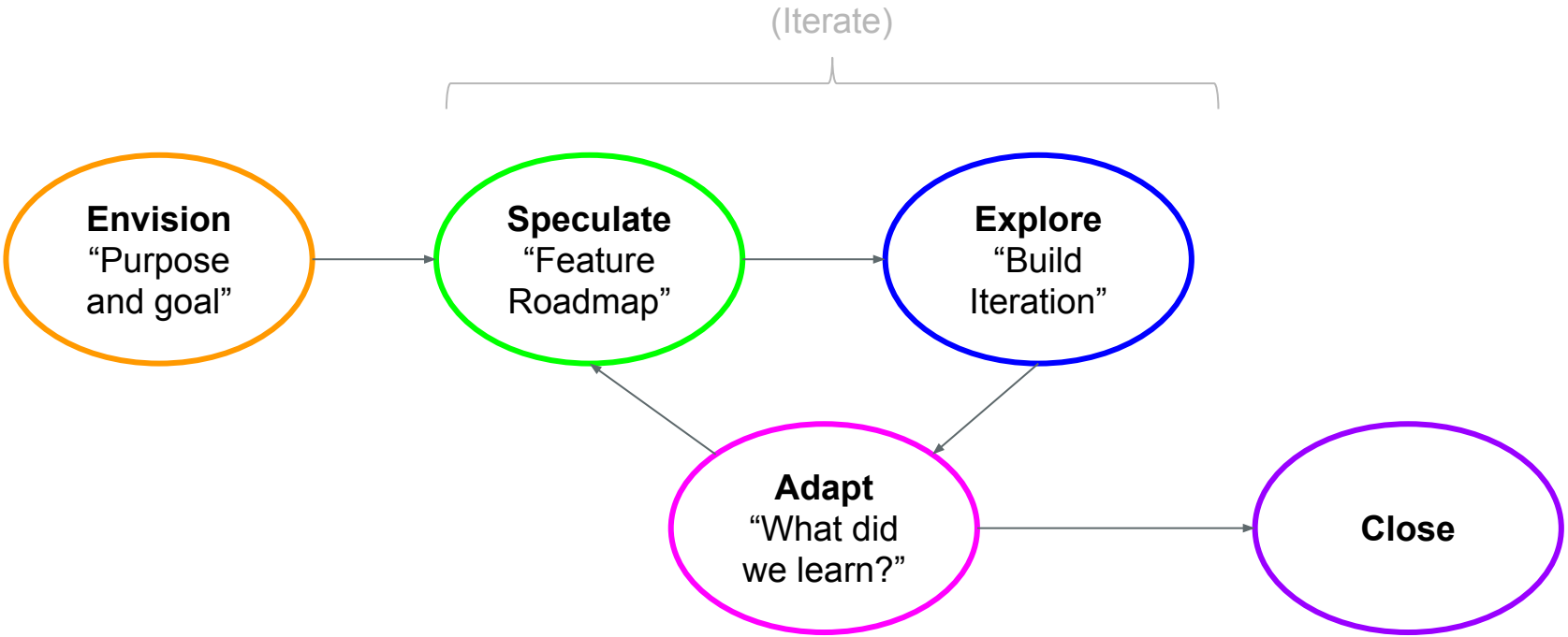
Remember, a set of people coming from different backgrounds were developing their own methodologies. They came together to summarize/distill their common beliefs on how to produce software.

Many process/techniques paper already in existence. However, they coined the movement.

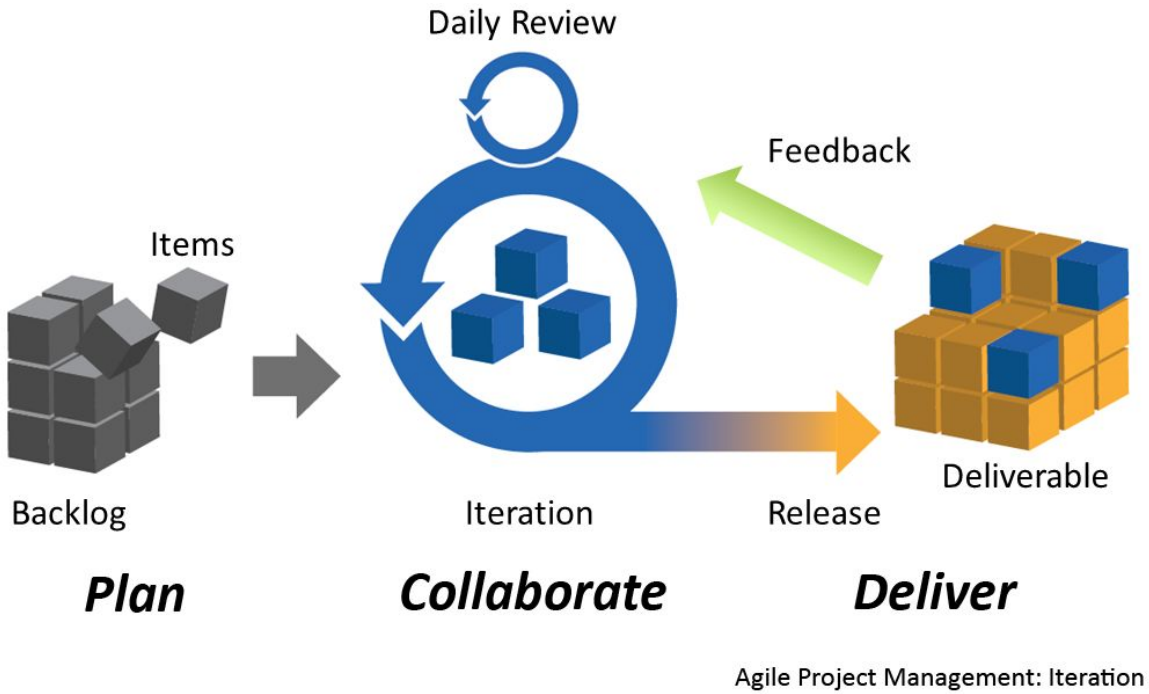
Agile Project Management: Agile Triangle



Agile Process: APM



Agile Process: APM



https://en.wikipedia.org/wiki/Iterative_and_incremental_development

Comparing Adaptive vs Predictive

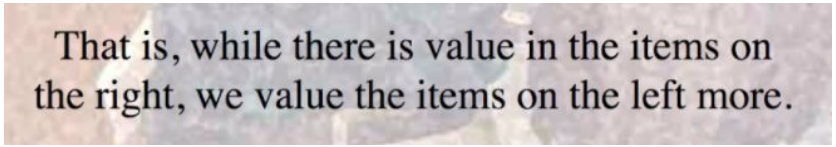
Comparing Predictive vs Adaptive

Aspect	Predictive	Adaptive (Agile)
Philosophy	Top down/predictive planning is possible and process driven	Individuals, Collaboration, Respond to Change, Working Software
Authority	Centralized (Process Maker)	Decentralized (Team)
Process	Sequential	Iterative
People vs Process	Process Oriented	People Oriented
Design	During design phase	Ongoing and evolvng
Feedback	Delayed	Rapid
Partnership	Contractual	Shared Success / Collaborative
Success	“On scope, time and budget”	“Deliver value”
Response to Change	Make a plan and resist change	Have a plan but embrace change

Be Careful I

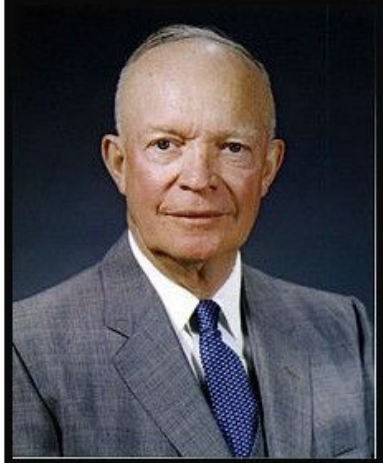
There is something subtle and important in the manifesto:

*"Responding to change **over** a plan"*



Does the manifesto recommend?

1. No Plan / Plans are Useless
2. Have a plan but adapt to change
3. Create a plan and it drives everything

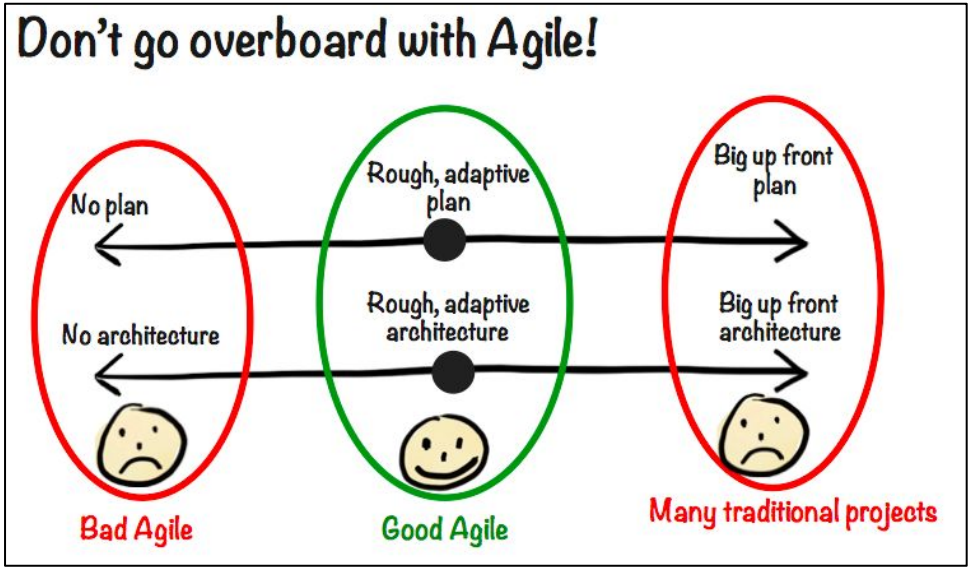


In preparing for battle, I have always found that plans are useless but planning is indispensable.

(Dwight D. Eisenhower)

izquotes.com

Be Careful II



Bad Agile

Cowboys use 'agile' as cover for **chaos**
Chaos = lack of plan and architecture

Q: "Where is the plan or design?"
A: "oh we are an agile team"

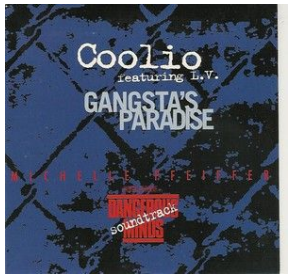
Predictive vs Adaptive vs Chaotic

Aspect	Predictive	Adaptive (Agile)	Chaos
Philosophy	Top down/predictive planning is possible and process driven	Individuals, Collaboration, Respond to Change, Working Software	Wake up in the morning and do whatever I want.
Authority	Centralized (Process Maker)	Decentralized (Team)	Individual
Process	Sequential	Iterative	Process? Anarchy
People vs Process	Process Oriented	People Oriented	Anarchy
Design	During design phase	Ongoing and evolving	What design?
Feedback	Delayed	Rapid	Ad-hoc
Partnership	Contractual	Shared Success / Collaborative	If you can work out how to engage the team?
Success	“On scope, time and budget”	“Deliver value”	Maybe?
Response to Change	Resist	Embrace	There was a plan?

Did it get any better?

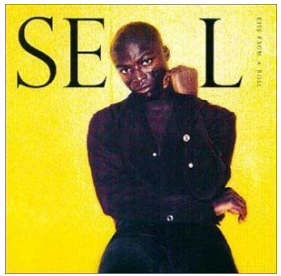
1995

Billboard top 5 included



“Gangstas Paradies”

“Waterfalls”



“Kiss from a rose”

1995 Billboard

What about the *software crisis*?

Standish CHAOS Report **2011-2015** covering 10k Applications

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

The resolution of all software projects from FY2011–2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.

https://www.infoq.com/articles/standish-chaos-2015

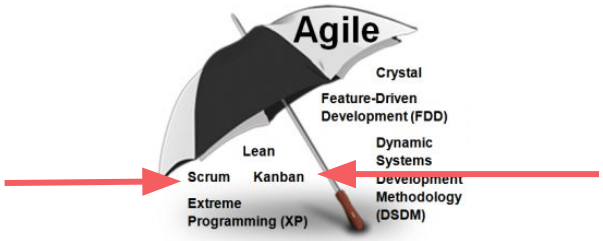
Methodology vs Methods

But Wait ....

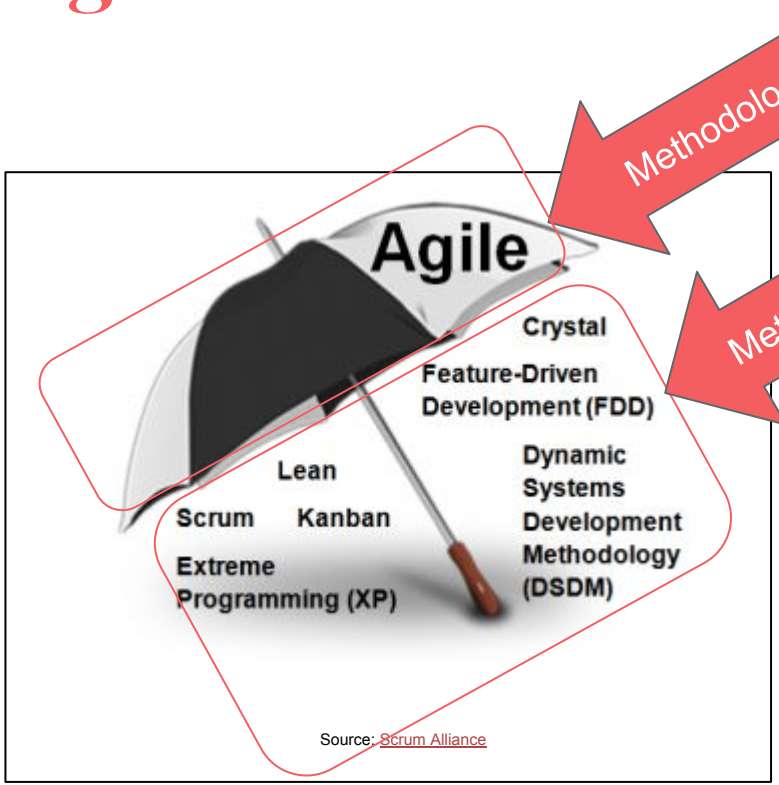
"I think I understand predictive, adaptive and agile
(though I will still do the reading!)

However, I still don't know how to apply this to my team / project?
This does not feel 'actionable'"

Next Lecture:
We will pick two agile frameworks - Scrum and Kanban and walk through



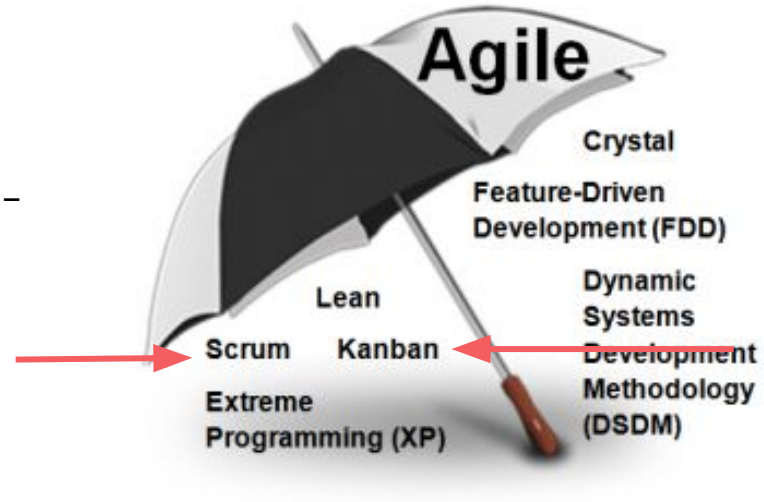
Agile Umbrella



Agile is an “umbrella term” to describe ***a set*** of software development **methods and practices** ***based on the values and principles*** of the Agile Manifesto

Next Lecture

We will pick two agile methods - Scrum and Kanban and walk through



Pop Quiz

Question	Answer
Predictive processes are characterized by _____?	
Agile is a concrete checklist to follow? True/False	
The key principles in the agile manifesto are _____?	
Agile just means “iterating”? True/False	
Agile defines an actionable set of steps for my team to follow?	
There are many processes/frameworks that implement Agile?	
Agile is _____ whereas Waterfall is _____ and ‘No Process’ is _____. (Choose from predictive, adaptive, chaotic)	

Reading

Reading	Optionality
Fowler on Agile	Required
Agile Alliance	Required
Beginning Software Engineering chap 12	Required
Agile Fluency	Required
The famous Agile Car (and the real meaning)	Required

Aside: Compare Agile Principles to the Post Mortem of Failed Project .....

Success Criteria	Points	DMV	CONFIRM	HYATT	ITAMARATI
1. User Involvement	19	NO (0)	NO (0)	YES (19)	YES (19)
2. Executive Management Support	16	NO (0)	YES (16)	YES (16)	YES (16)
3. Clear Statement of Requirements	15	NO (0)	NO (0)	YES (15)	NO (0)
4. Proper Planning	11	NO (0)	NO (0)	YES (11)	YES (11)
5. Realistic Expectations	10	YES (10)	YES (10)	YES (10)	YES (10)
6. Smaller Project Milestones	9	NO (0)	NO (0)	YES (9)	YES (9)
7. Competent Staff	8	NO (0)	NO (0)	YES (8)	YES (8)
8. Ownership	6	NO (0)	NO (0)	YES (6)	YES (6)
9. Clear Vision & Objectives	3	NO (0)	NO (0)	YES (3)	YES (3)
10. Hard-Working, Focused Staff	3	NO (0)	YES (3)	YES (3)	YES (3)
TOTAL	100	10	29	100	85

Source: Standish Group Chaos Report 1994, Other sources: IEEE, Surveys