Agile Software Development with Scrum

Jeff Sutherland
Gabrielle Benefield

Scrum Training Institute

Agenda

Introduction

- Overview of Methodologies
- Exercise; empirical learning
- Agile Manifesto
- Agile Values
- History of Scrum
- Exercise: The offsite customer

Scrum 101

- Scrum Overview
- Roles and responsibilities
- Scrum team
- Product Owner
- ScrumMaster

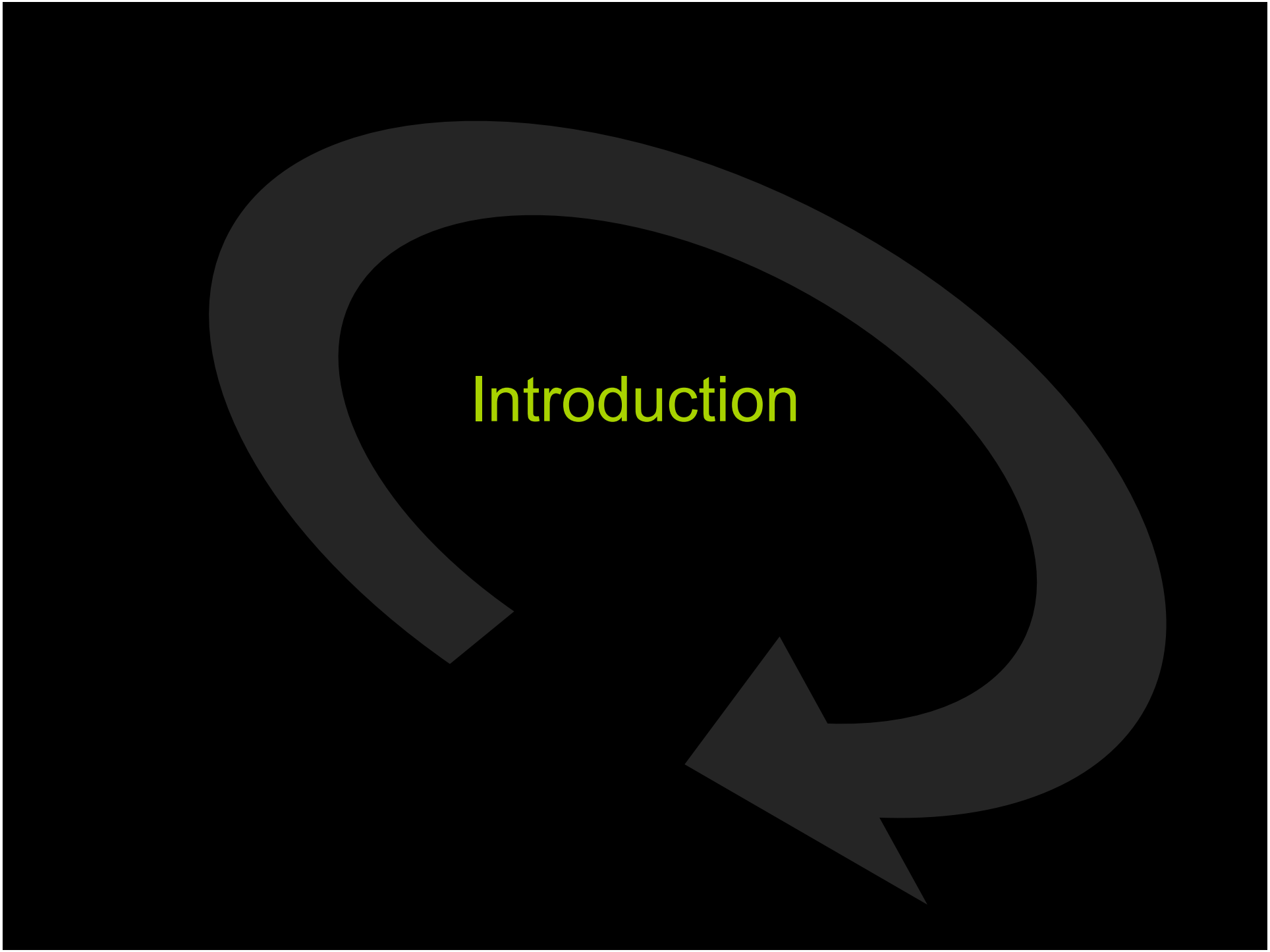
Agenda

Scrum In-depth

- The Sprint
- Sprint Planning
- Exercise: Sprint Planning
- Tracking Progress
- Daily Scrum
- Exercise: Running a Daily meeting
- Sprint Review
- Sprint Retrospective
- The ScrumMaster role:
- Exercise: Great ScrumMaster scenarios
- Exercise: Velocity game

Optional modules:

- Scaling Scrum
- Distributed Scrum

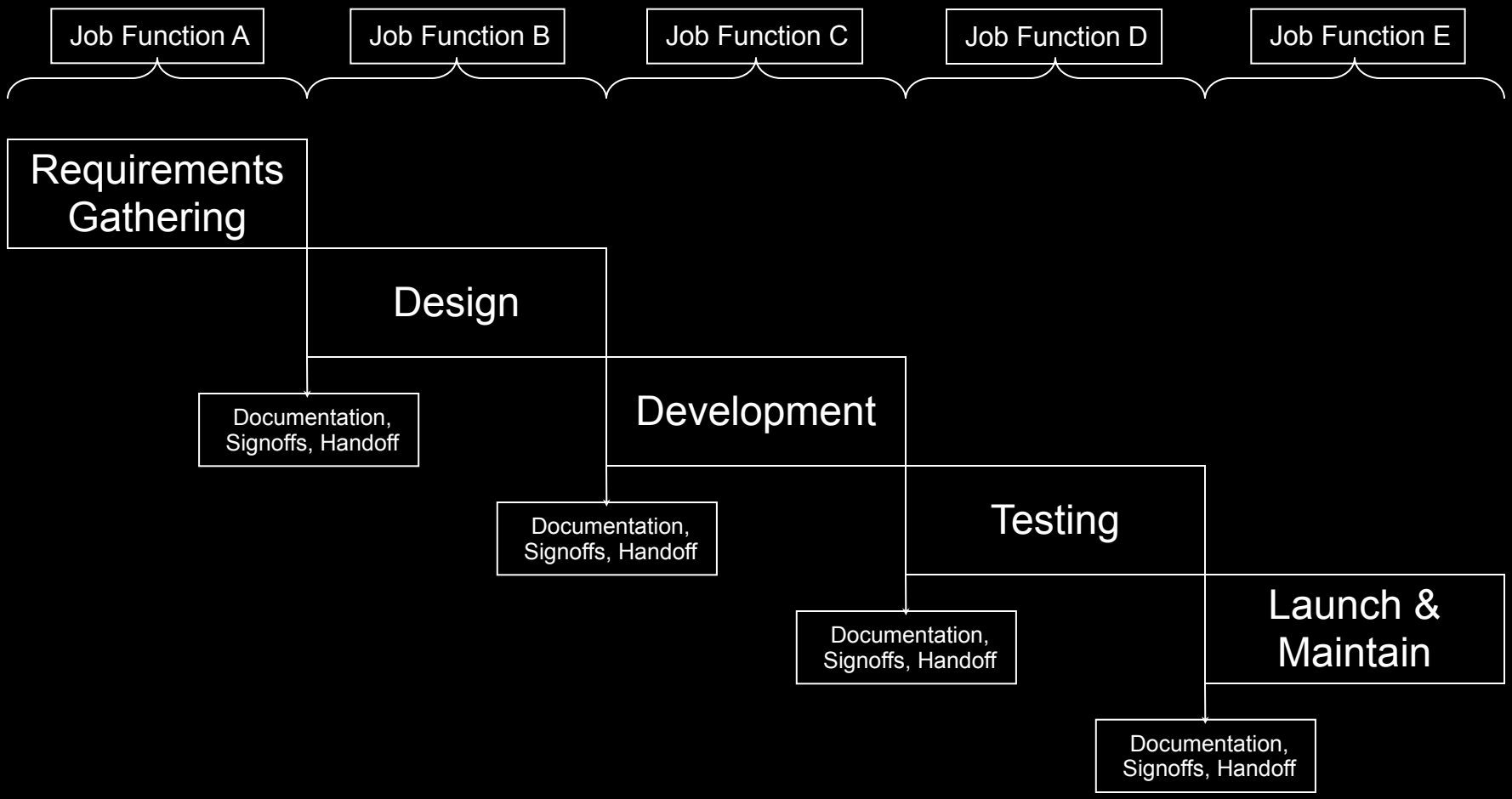

Introduction

A Defined Process

Donut Mix

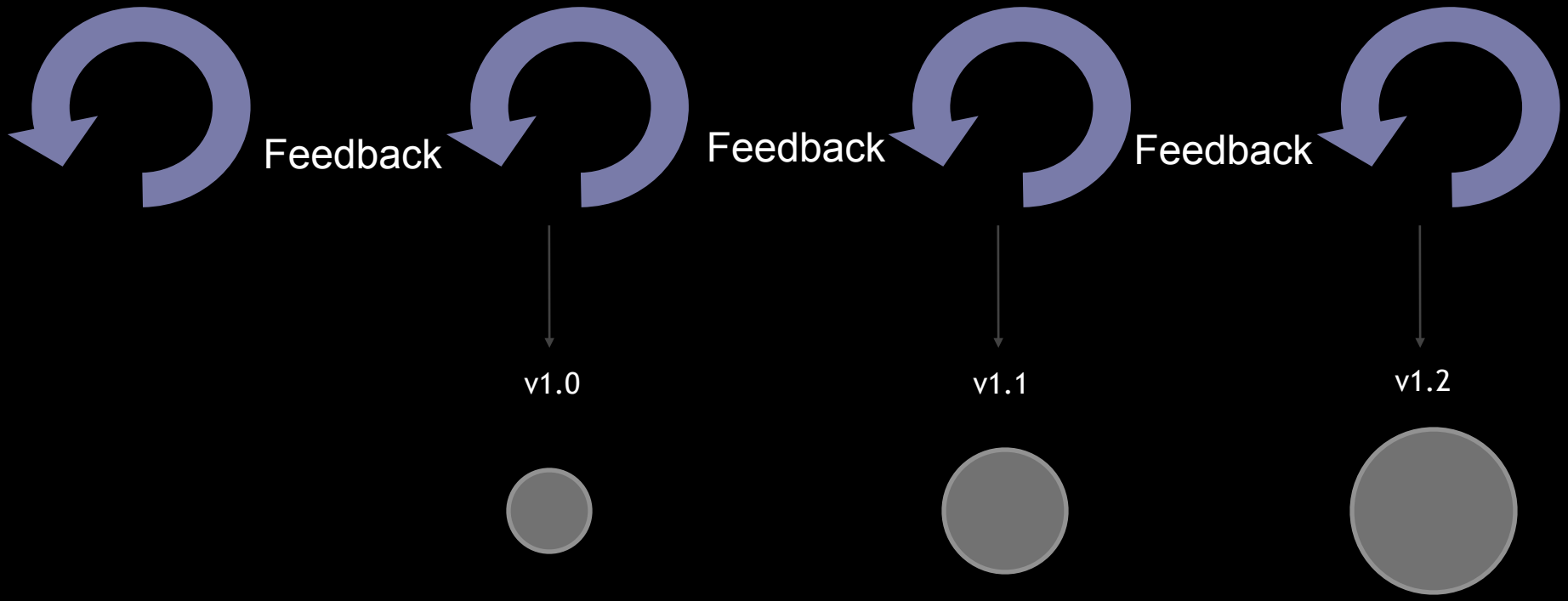
Yummy Donuts!

The Waterfall



Advantage: Highly Logical
Disadvantage: High failure rate, low process efficiency

Agile Development



- Short Iterations of 1-4 weeks
- Incremental releases

Agile Development

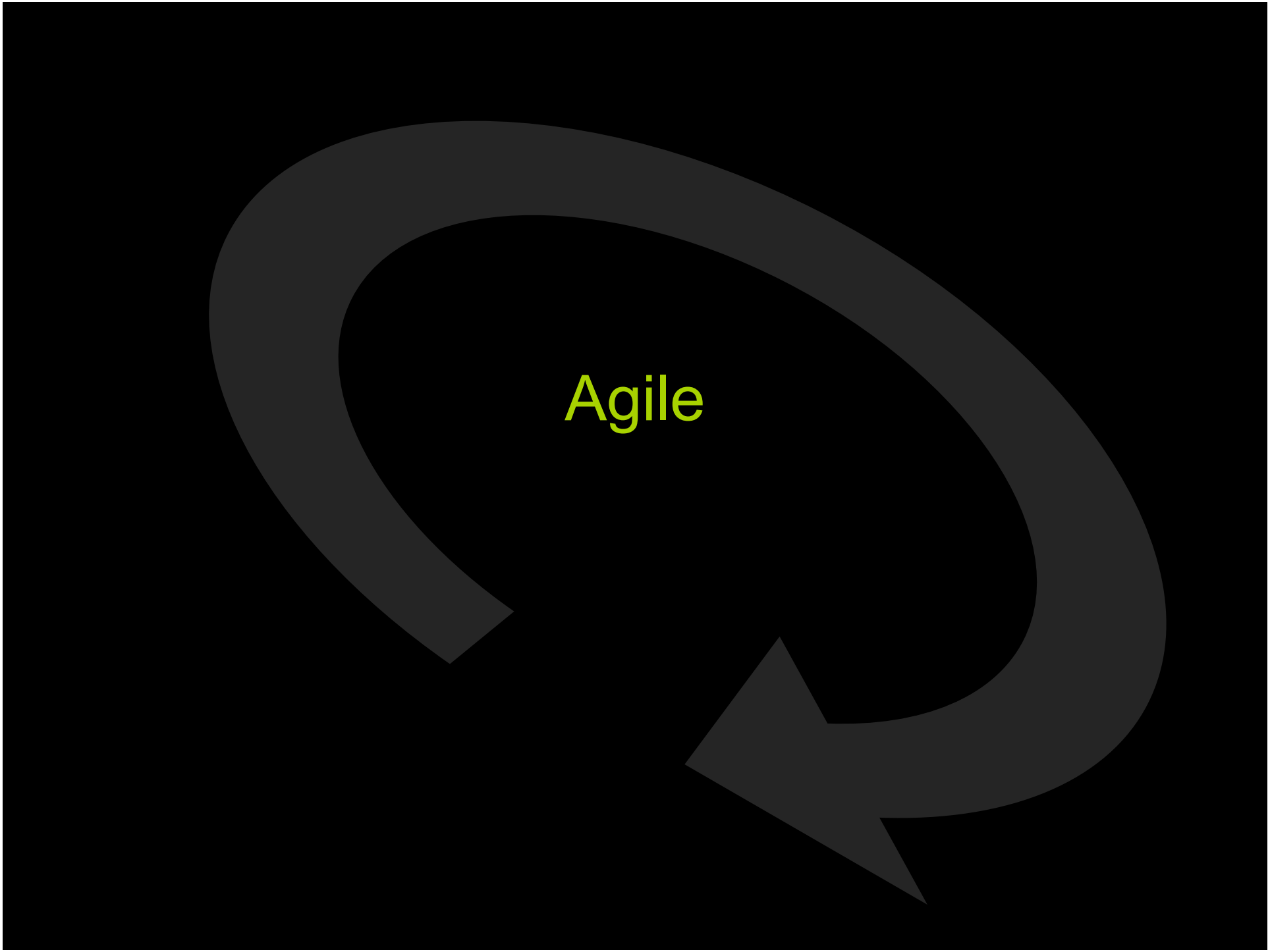
- Plan
- Test
- Design
- Build

- Plan
- Test
- Design
- Build

- Plan
- Test
- Design
- Build

- Plan
- Test
- Design
- Build

- Do a little bit of everything every cycle


Agile

“Agile” is a great buzzword. Who doesn’t want to be Agile?

No one says, “Thanks, I’d rather be inflexible and slow to respond.”

The Agile Manifesto – 2001

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

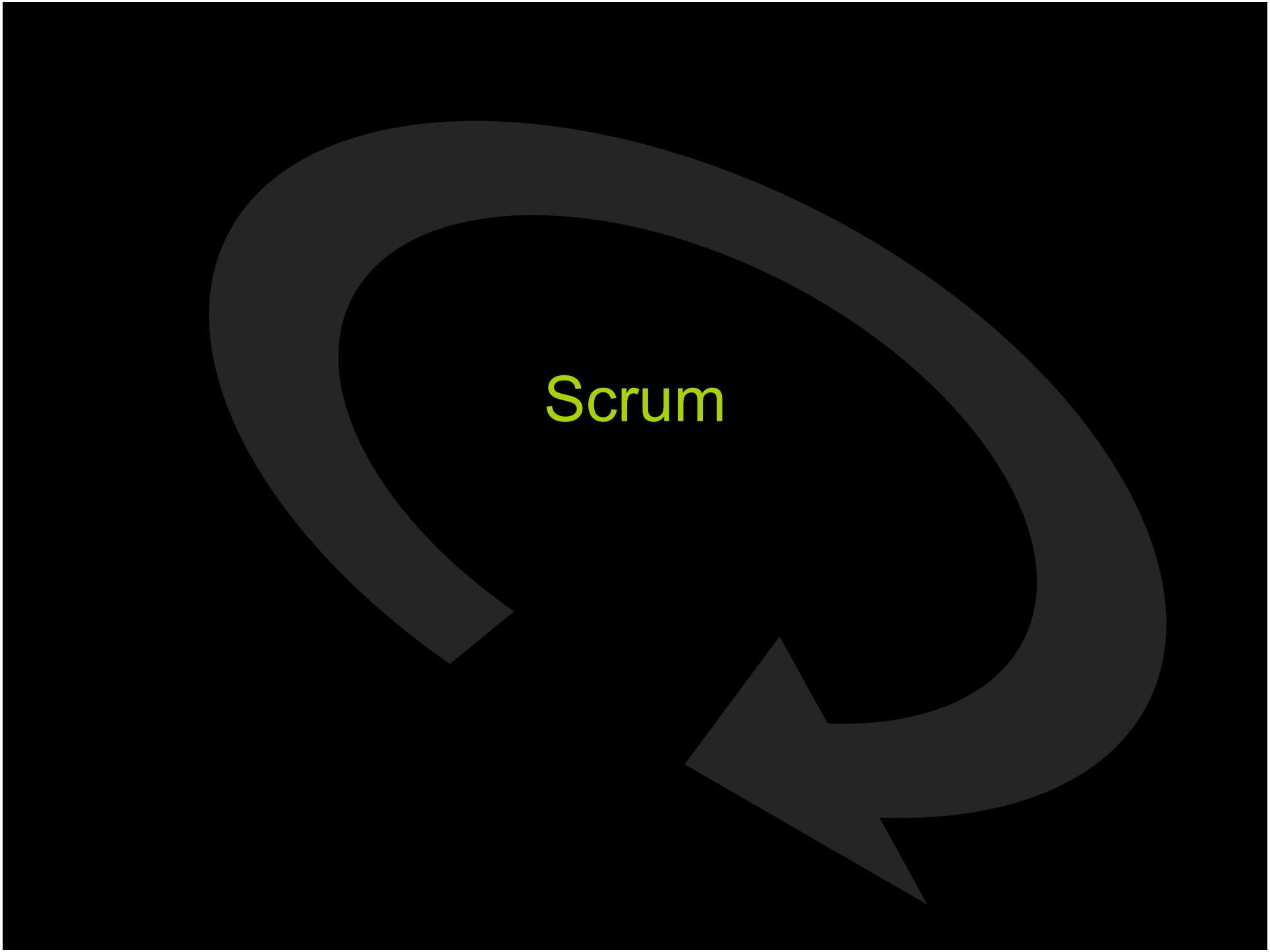
That is, while there is value in the items on the right,

we value the items on the left more.

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

Agile Myths

- No design
- No planning
- Poor quality
- “Cowboy” Coding
- Only works on small projects


Scrum

What is Scrum?

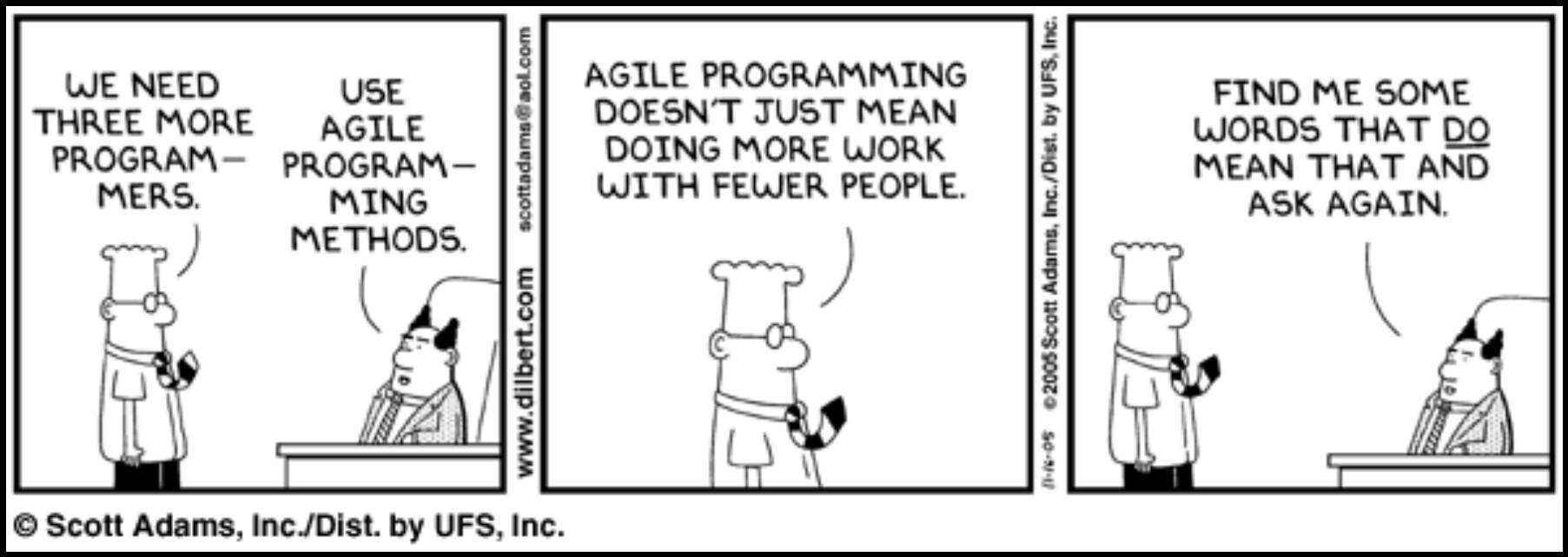
A flexible framework that is:

- Collaborative
- Iterative & Incremental
- Commonsense
- Very simple but very hard; it causes change.

Scrum Values

- Respect
- Commitment
- Focus
- Openness
- Courage

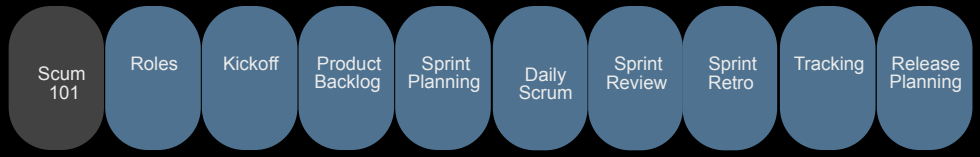
A silver bullet?



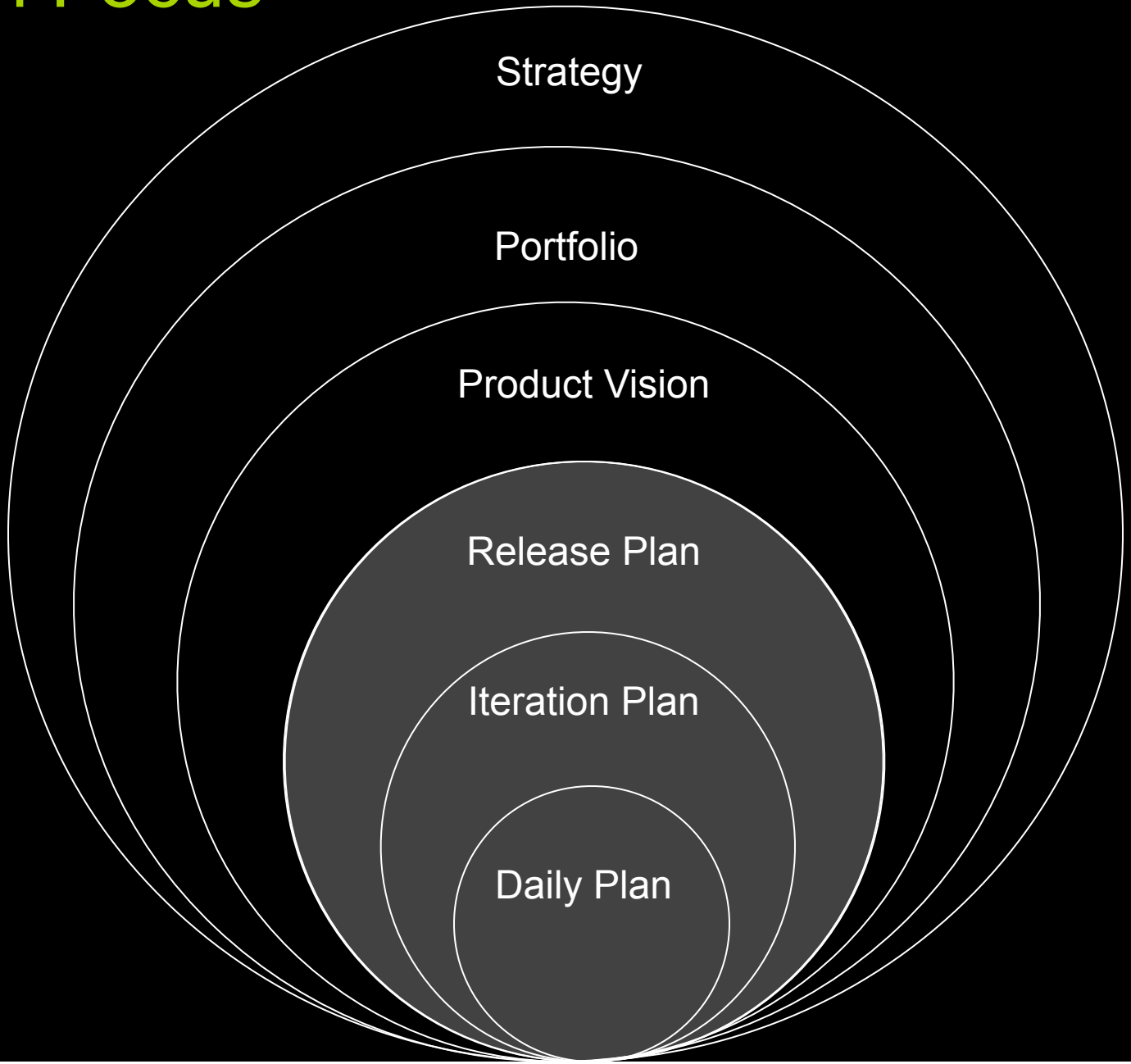
Scrum Disadvantages

- Makes all dysfunction visible
- Bad products will be delivered sooner, and doomed projects will fail faster
- People follow the mechanics but not the values.

Scrum 101


Scum 101
Roles
Kickoff
Product Backlog
Sprint Planning
Daily Scrum
Sprint Review
Sprint Retro
Tracking
Release Planning

Scrum Focus


Strategy
Portfolio
Product Vision
Release Plan
Iteration Plan
Daily Plan

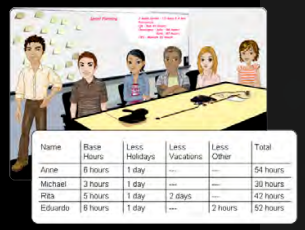
Scrum 101

Daily Stand-up

Sprint Review

Potentially Shippable Product Increment

Sprint Planning



Sprint

Product Backlog

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	

Commitment

No Changes
(in Duration or Deliverable)

Team Retrospective

Challenges
- Testing
- Build breaks

Improvement

Scrum Roles

Product Owner

Determines what should be produced, to maximize ROI for the business

The Team

Cross-functional and self-organizing team of 7 people +/- 2

ScrumMaster

Protects and serves the Team, and supports their practice of Scrum

Product Backlog

This is the Product Backlog

- A prioritized list of requirements
- Prioritized by the product owner

1
2
3
4
5
6
7
8
9
10
11
12
13

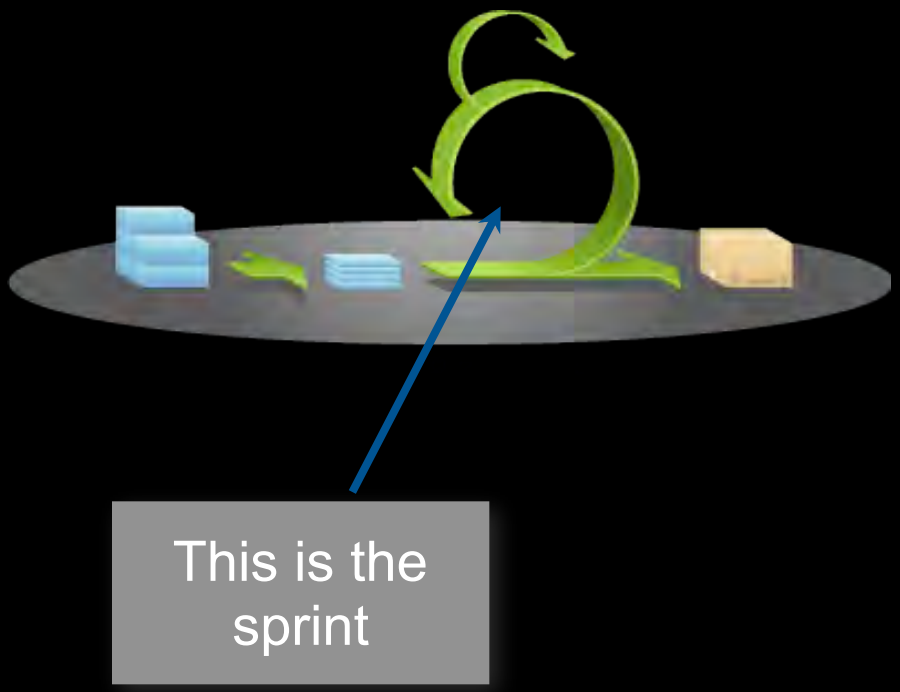
Sprint Planning

This is the Sprint Planning Meeting

- Planning at the start of a Sprint by the whole team and Product Owner
- Team creates tasks, estimates, and volunteers for them

Name	Base Hours	Less Holidays	Less Vacations	Less Other	Total
Anne	6 hours	1 day	---	---	54 hours
Michael	3 hours	1 day	---	---	30 hours
Rita	5 hours	1 day	2 days	---	42 hours
Eduardo	8 hours	1 day	---	2 hours	52 hours

Sprints

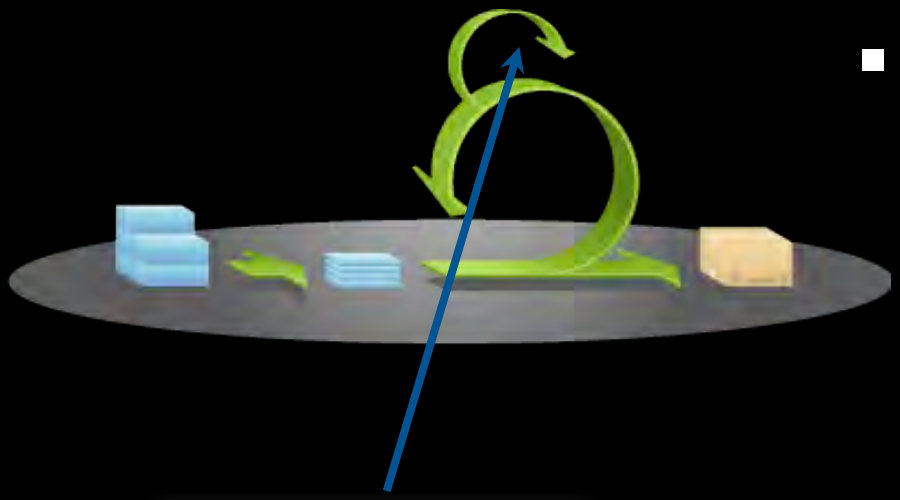


1 month timebox of work
During the Sprint:

- Analysis
- Design
- Code
- Test

A little bit of everything

Daily Stand-Up



This is the Daily Meeting (Stand-Up)

- A daily team meeting
- Keep up to date
- Help each other to resolve problems

Sprint Review

A demo by the team of:

- Complete
- Fully tested
- Potentially shippable features

Anyone can attend

Sprint Retrospective

This is the
Sprint Retrospective

- A meeting at the end of each Sprint so the team can Inspect and Adapt the process.

Two week Sprint Cycle Example

First Sprint

M	T	W	Th	F
~ 9:00 - 12:00 **Sprint Planning** 	Daily Scrum 	Daily Scrum 	Daily Scrum 	Daily Scrum 
Daily Scrum 	Daily Scrum 	Daily Scrum 	Daily Scrum 	**2-3pm Sprint Review**  **3.30- 4.30pm** **Sprint Retrospective**

Second Sprint

~ 9:00 - 12:00 **Sprint Planning** 	Daily Scrum 	**Etc…**		

Are you doing Scrum?

1. Do you deliver working software at the end of each Sprint (less than 4 weeks) that is tested at the feature level?

2. Do you do just enough specification before starting a Sprint and are their near-term Product Backlog items small and clearly understood to be completed within a sprint?

3. Do you have a Product Owner and a Product Backlog with prioritized items estimated by the team?

4. Does the team have a release burndown chart and does the team know their velocity?

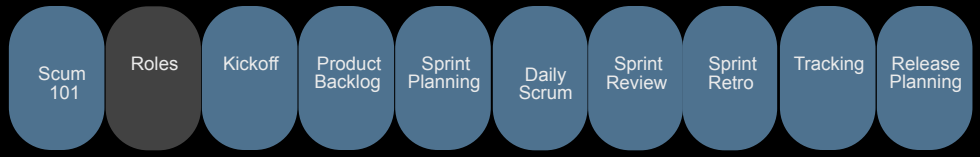
You know when you are not doing iterative development when:

- Iterations are longer than 2-4 weeks
- Team tries to complete specification before programming
- An iteration does not include testing
- Iteration does not produce workable code
- Detailed (task level) plans are accurate estimates and expected at the beginning of a project
- The sprint plan doesn’t reflect what the team is doing.

You know when you are not doing agile development when:

- There is little co-operation within the team
- Design and code is produced by individual effort
- Progress is measures by hours spend or documents created instead of working code
- Builds are done once every three weeks.

Roles


Scum 101
Roles
Kickoff
Product Backlog
Sprint Planning
Daily Scrum
Sprint Review
Sprint Retro
Tracking
Release Planning

Product Owner

- Product visionary
- Maximizes business value
- Prioritizes and clarifies
- requirements.

Product Owner Responsibilities

Does

- Provide clear product direction
- Work with the team closely to clarify requirements
- Actively manages the product backlog
- Represents the business and customer needs.

Does not

- Assign work to team members
- Give fixed date fixed scope projects without team consent
- Change priorities during an iteration.

Exercise: Choosing a Product Owner

- In your team come up with a list of attributes for your Product Owner
- Discuss who would make a good Product Owner in your organization

The Scrum Master

- Similar to a Project Manager,
  - …yet different
- A facilitator
- Removes impediments

The Role of the ScrumMaster

- The ScrumMaster does everything in their power to help the team achieve success
- This includes:
  - Serving the team
  - Protecting the team
  - Guiding the team's use of Scrum

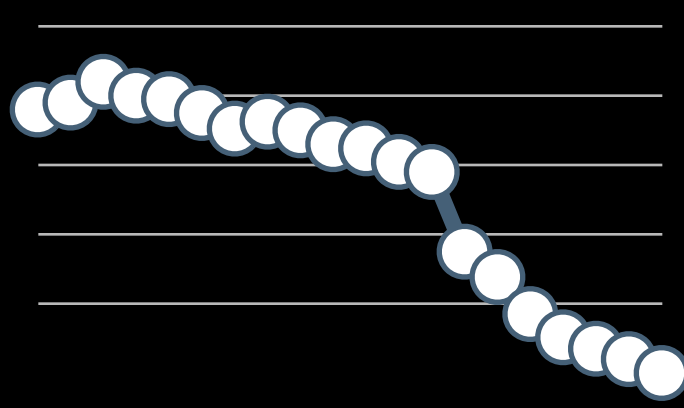
What does a ScrumMaster do?

- Setup meetings
- Facilitate meetings

M	T	W	T	F
~ 9:00 - 12:00 Sprint Planning 	Daily Scrum 	Daily Scrum 	Daily Scrum 	Daily Scrum 
Daily Scrum 	Daily Scrum 	Daily Scrum 	Daily Scruma 	2-3pm Sprint Review 3.30- 4pm Sprint Retrospective

What do they do?

- Coach team in Scrum values and practices
- Protect from disruption
- Remove impediments
- Listen and help
- Track progress



Exercise: Facilitation

- Work with your table to describe the difference between “directing” and “facilitating”
  - Directive: “Do this task now”
  - Facilitative: “What support do you need to get around this problem”?
- Working with your table, come up with 2 “directive” and 2 “facilitative” statements.

Exercise: Choosing a ScrumMaster

- In your team come up with a list of attributes for a good ScrumMaster
- Discuss who could be a ScrumMaster in your organization.

The Scrum Team

Cross-functional
Self-Organizing
Shared Commitment.

Team

- 7 people, + or – 2
  - Has worked with as high as 15, as few as 3
  - Can be shared with other teams (but better when not)
  - Can change between Sprints (but better when they don’t)
  - Can be distributed (but better when colocated)
- Cross-functional
  - Possesses all the skills necessary to produce an increment of potentially shippable product
  - Team takes on tasks based on skills, not just official “role”
- Self-managing
  - Team manages itself to achieve the Sprint commitment

Basic truths about team motivation

- People are most productive when they manage themselves;
- People take their commitment more seriously than other people's commitment for them
- People do their best work when they aren't interrupted
- Teams improve most when they solve their own problems.

Ideal Environment

- Everyone in same location
- Open space without barriers
- Resources available; phone, whiteboards, meeting space etc

Collaborative workspace

What happens to the manager in Scrum?

Manager 2.0

- Cross off the list everything that:
  - Conflicts with Scrum
  - Is unnecessary in Scrum
  - Would undermine the team's self-organization and self-management

Decide task assignments among the team members and assign them	Provide coaching and mentorship to team-members
Keep track of whether team-members have done the tasks I've assigned to them	Surface issues to the team that they might overlook – scaling, performance, security, etc.
Make commitments on behalf of the team about how much they can get done by X date	Provide input on features, functionality, and other aspects of what's being produced
Give direction to the team on how to do the work, so they can meet the commitment I made	Do performance evaluations and provide feedback to team-members
Convince team that the commitments made on their behalf are attainable	Provide advice and input to the team on difficult technical issues that come up
Monitor the team's progress, to make sure they stay on schedule, and aren't having problems	Plan training for team, and do career-development and planning with team-members
Conduct weekly update and 1:1 meetings with team, to surface issues, and provide direction	Stay abreast of latest developments in the technology their team uses, industry news, etc.
Recruit, interview and hire new members of the team	Plan and oversee budgets and financials, and think about tools, skills and other future needs
Fire team-members who are consistently not able to perform	Help remove impediments that the team is not able or well-placed to resolve themselves