

Scrum, but

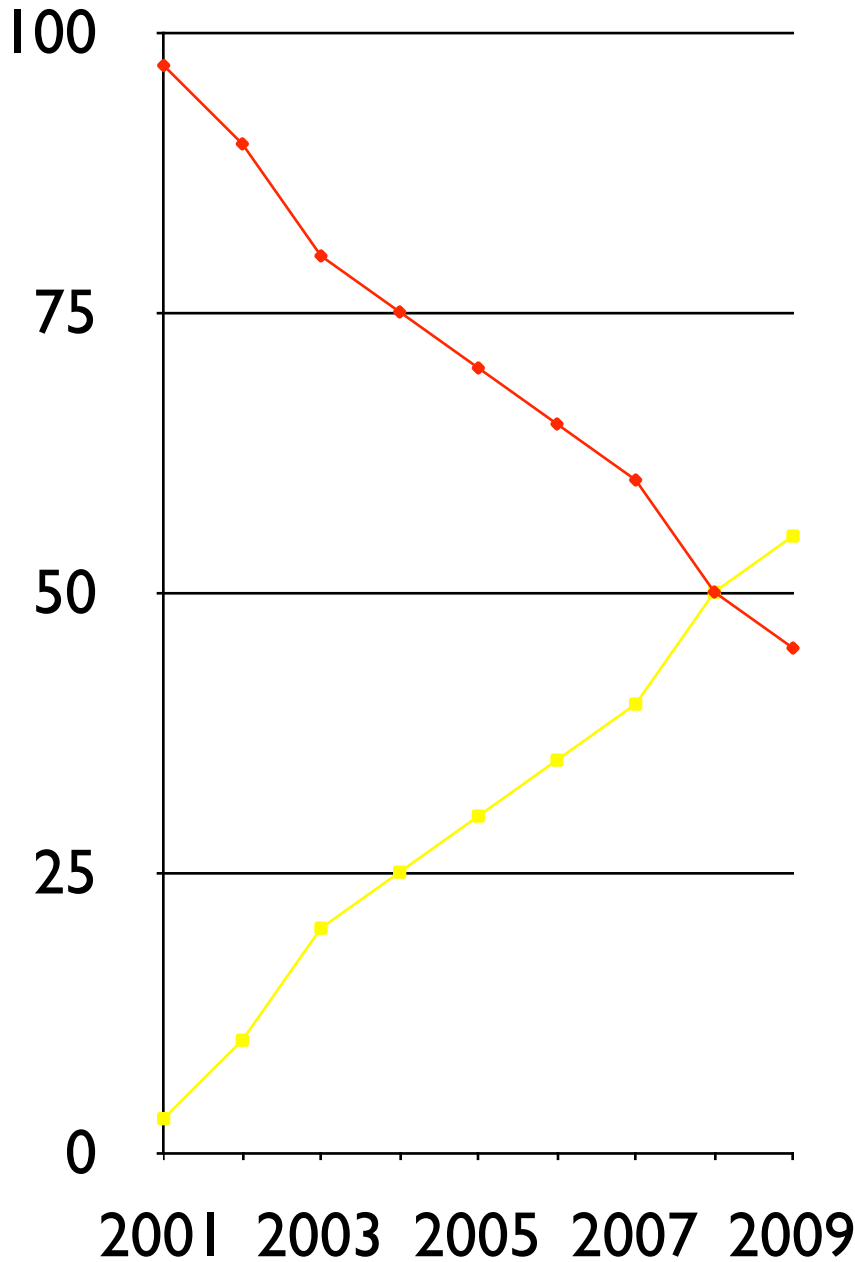


ScrumAllianceSM
transforming the world of work

Waterfall %

Agile %

2

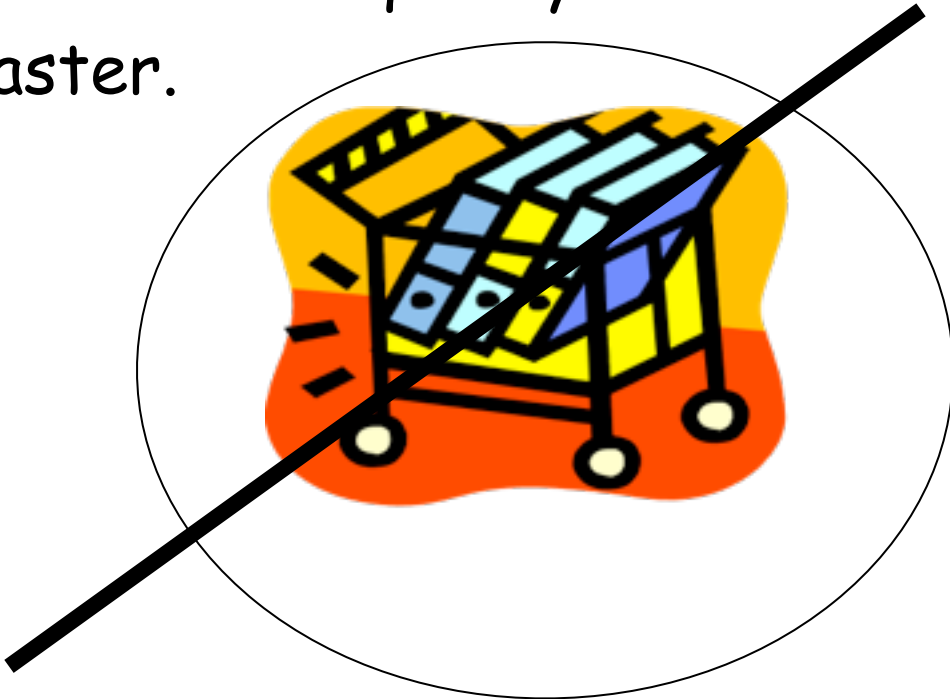


In 2008, 84% of all Agile projects used Scrum

Source: December 2008 Global Agile Company Online Survey

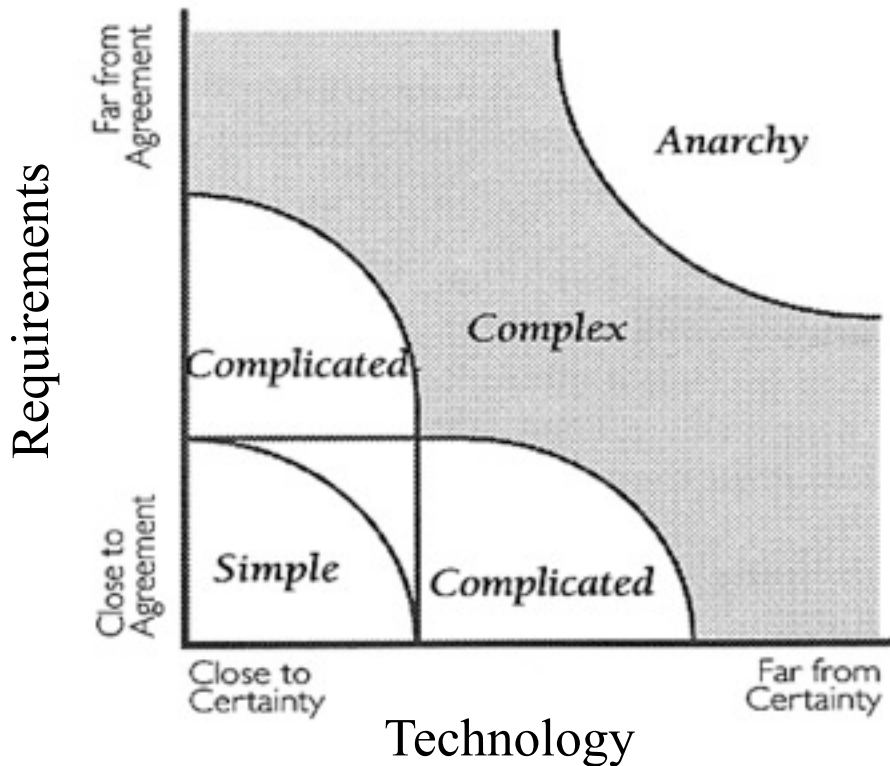
Introduction and Basics

Scrum is not a methodology that will make you develop better products. Scrum does not provide the answers to how to build quality software faster.



Scrum is a tool, a framework, you can use to find out what you need to do to build quality software faster.

Categorization of complexity in development projects




- People dimension adds another level of complexity

Type	Characteristics	Leader's Job
Chaotic	<p>High Turbulence</p> <p>No clear cause-and-effect</p> <p>Unknowables</p> <p>Many decisions and no time</p>	<p>Immediate action to re-establish order</p> <p>Prioritize and select actionable work</p> <p>Look for what works rather than perfection</p> <p>Act, sense, respond</p>
Complex	<p>More unpredictability than predictability</p> <p>Emergent answers</p> <p>Many competing ideas</p>	<p>Create bounded environments for action</p> <p>Increase levels of interaction and communication</p> <p>Servant leadership</p> <p>Generate ideas</p> <p>Probe, sense, respond</p>
Complicated	<p>More predictability than unpredictability</p> <p>Fact-based management</p> <p>Experts work out wrinkles</p>	<p>Utilize experts to gain insights</p> <p>Use metrics to gain control</p> <p>Sense, analyze, respond</p> <p>Command and control</p>
Simple	<p>Repeating patterns and consistent events</p> <p>Clear cause-and-effect</p> <p>Well establish knows</p> <p>Fact based management</p>	<p>Use best practices</p> <p>Extensive communication not necessary</p> <p>Establish patterns and optimize to them</p> <p>Command and control</p>

Defined, Predictive

Start
with
Plan and
all
require
ments

End with all
requirements
completed



Scrum - Empirical

Start
with
Goals
and
some
priority
require
ments

End with
Goals
met



Defined, Predictive

Start with Plan and all requirements → End with all requirements completed

Start with Goals and some priority requirements



End with Goals met

Scrum - Empirical

Scrum

Empirical process for managing the development and deployment of complex products.

Empiricism is dependent on frequent **inspection and adaptation** to reach goal.

Inspection is dependent on **transparency**.

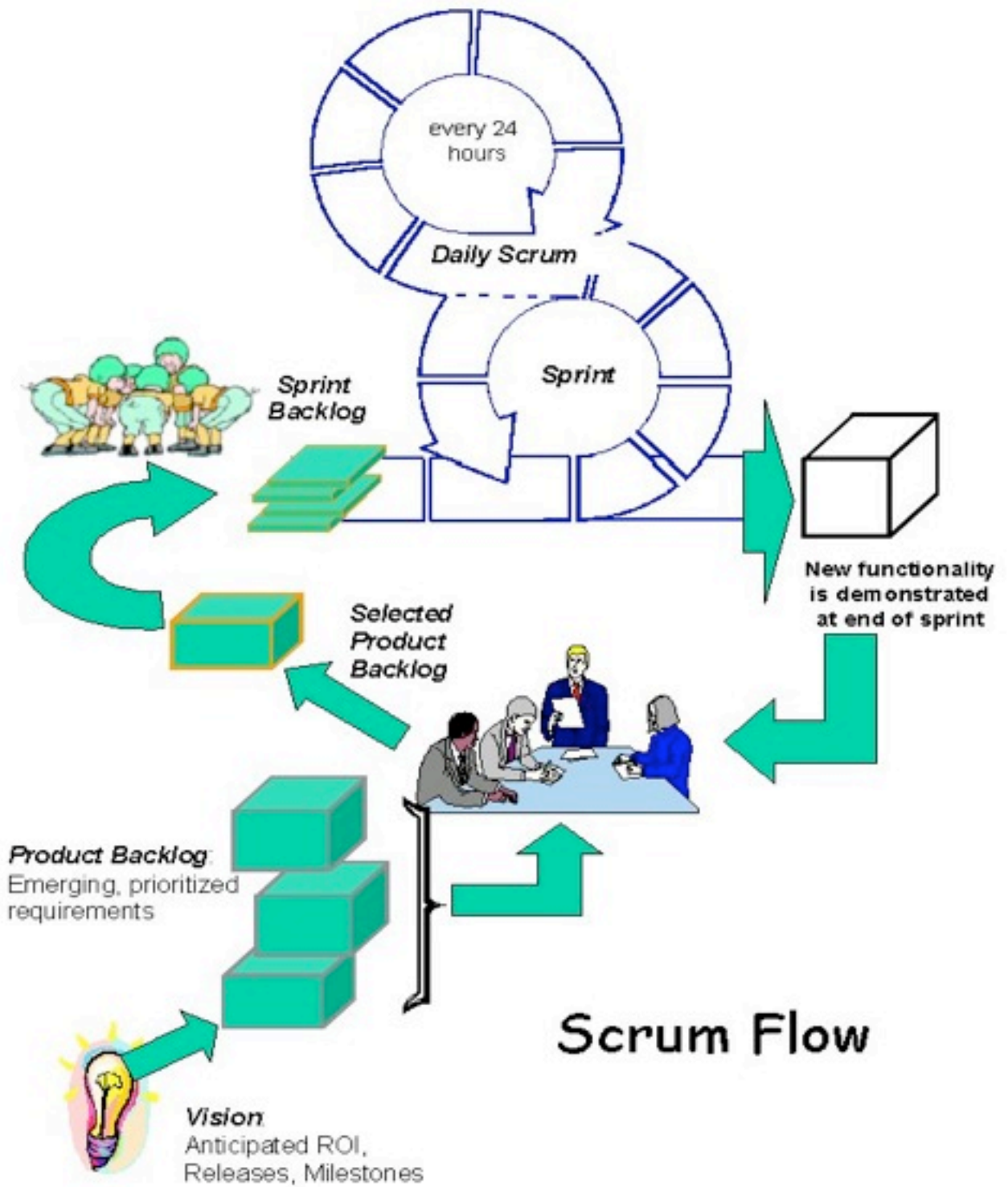
Scrum rests on the four legs of **iterative** development that generates done **increments** of functionality using **self-managing teams** that are **cross-functional**.

Empirical Process Control

- Uses Inspection and subsequent adaptation to optimize realization of goals.
- Transparency is required for inspection and adaptation.
- Transparency requires courage and change in reward systems.



© Scott Adams, Inc./Dist. by UFS, Inc.



Timeboxes, Roles, Rules

ScrumBut

Scrum is a tool that you can use to:

- Increase productivity
- Increase predictability
- Increase risk management capabilities
- Increase the value of products and systems
- Increase quality
- Improving the morale and pleasure of the developers, product managers, customers and stakeholders

Scrum's use in spreading in product companies and engineering organizations.

Many of them suffer from one or more of the following:

- Release schedules slipping
- Stabilization at end of release taking longer and longer
- Releases are taking longer and longer
- Planning seems to take too long
- Changes are hard to introduce mid-release
- Quality is deteriorating
- Death marches are hurting morale

These are not dysfunctions. They are manifestations of a smoothly operating organization.

If they choose to improve, they choose change.

If they choose to use Scrum as a tool to address these problems, they are choosing hard work.

Only 30% will succeed. The others will ScrumBut.

Mother-In-Law

The person who knew that her son/
daughter could have married
better, and who intends to help you
be good enough. You have just
invited her to come live with you.



However,

When they use Scrum, they run into ScrumButs.

ScrumButs are reasons why they can't take full advantage of Scrum to solve the problems and realize the benefits.

- Business doesn't want to be involved;
- Everyone wants their stuff first and can't agree;
- People aren't available to work on teams full-time;
- Teams don't see a need for a daily meeting
- Teams can't get a piece of functionality done in one Sprint;
- Teams don't have the skills to "do" something
- Teams can't fit regression and integration testing into the same Sprint as development;
- Teams aren't collocated - they are dispersed;
- ScrumMaster tells the team what to do and how to do it;
- Other managers, including functional managers, can't stay out of a Sprint;
- Important things come up that require interrupting the Sprint;
- The Sprints can't start until all of the other groups do their up-front work (usability, architecture, database, etc.);
- Other groups are building hardware or using waterfall.

A ScrumBut has a particular syntax:

(ScrumBut) (Reason)(Workaround)

An example ScrumBut is:

"We use Scrum, but (The Daily Scrum meetings are too much overhead) (so we only have them once a week, unless we need them more often)"

Every Scrum role, rule, and meeting is designed to provide the desired benefits and address the problems.

ScrumButs mean that Scrum has exposed a dysfunction that is contributing to the problem, but is too hard to fix. A ScrumBut retains the problem while modifying Scrum to make it invisible. The dysfunction is no longer a thorn in the side.

For every ScrumBut, create a ScrumBut aargh, consisting of:

ScrumBut

Principle : (what Scrum principle is being "butted")

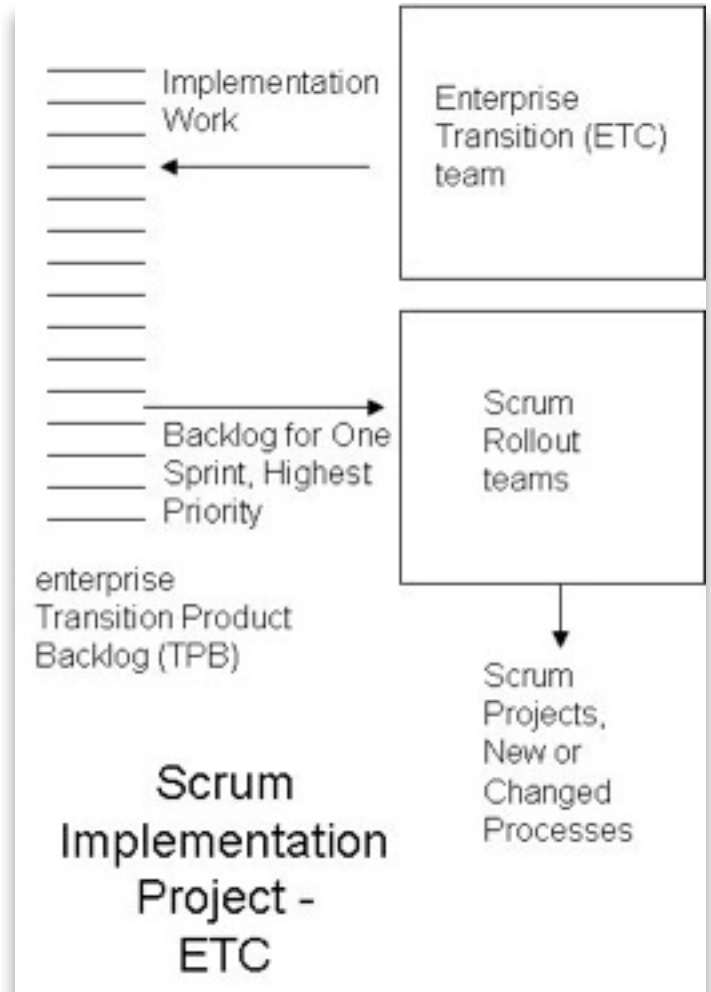
Consequence: (what potential benefit has been diminished) and (quantify the value diminished)

Use Scrum to Change

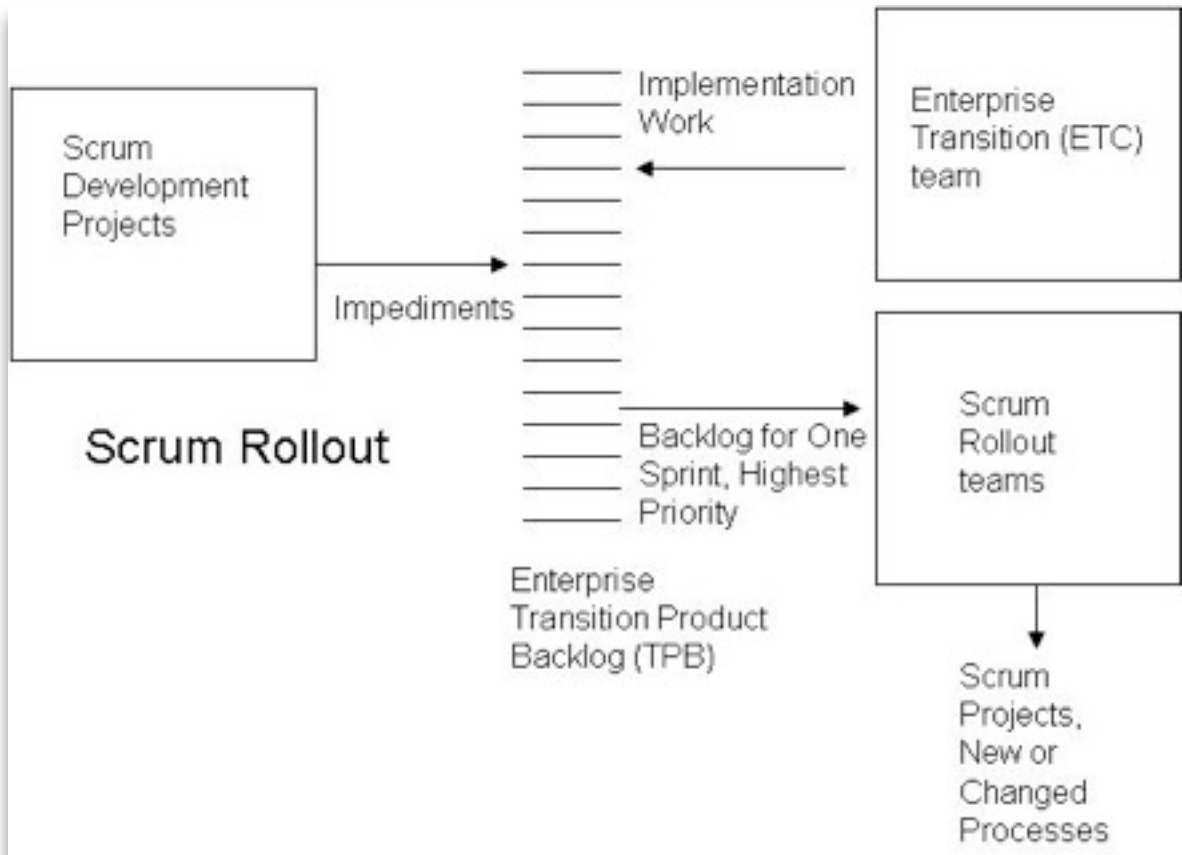
Enterprise Transition team (ETC) led by top person and his/her senior managers.

ETC uses Scrum and consist of a Product Owner, ScrumMaster, and team.

Changes made by Scrum Rollout teams.



The Whole Ball of Wax



ScrumButs

10/05/08

Copyright 2003-2008, ADM and MountainGoatSoftware, All Rights Reserved v8.3

ScrumBut 1

<We use Scrum, but>

<we can't get everything done in the development Sprint>

<so have had to form other Sprints and even assign people to work outside Scrum>

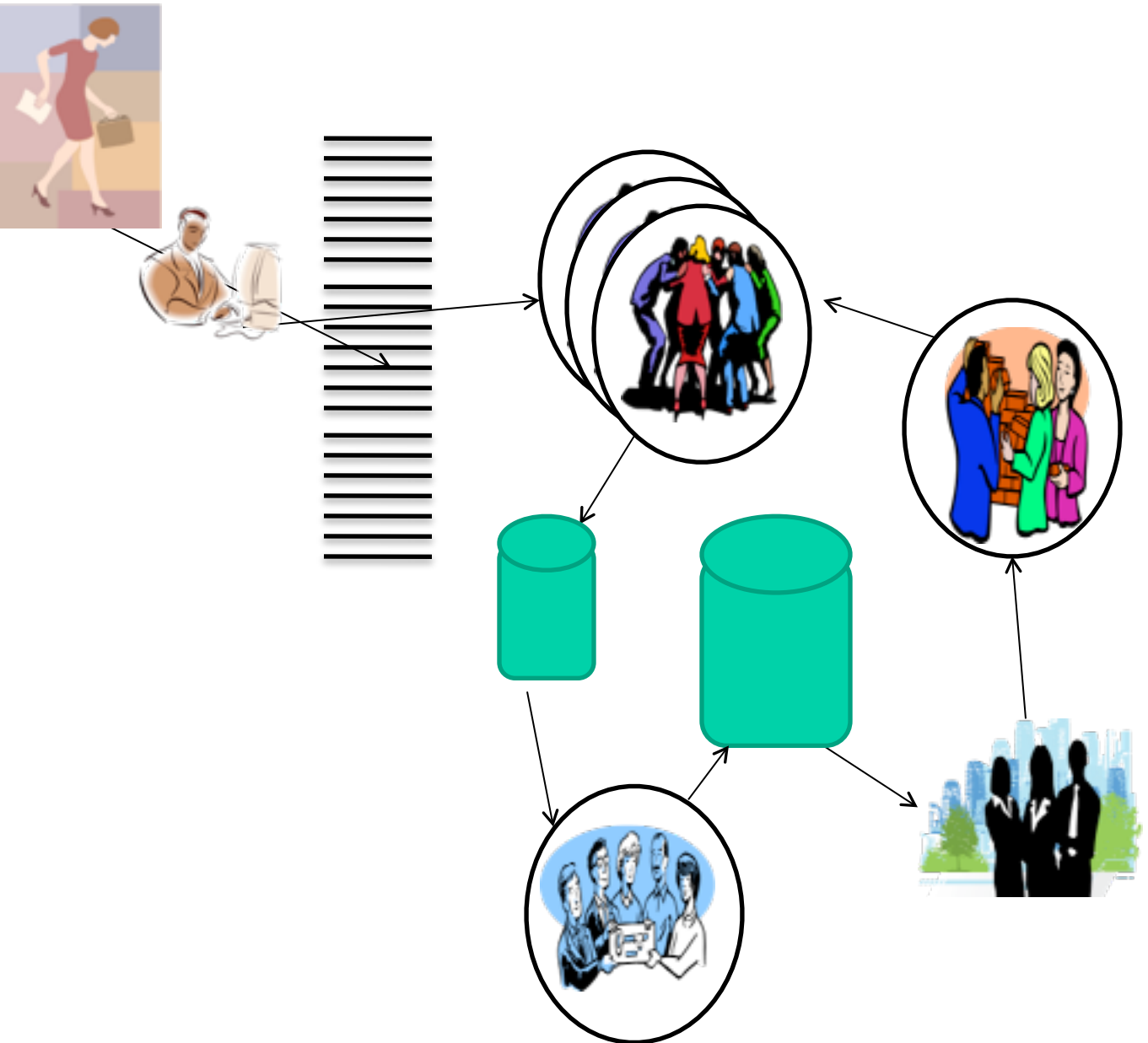
Principle - A complete product increment is done so all skills are applied, transparency exists, and interruptions are minimized.

Consequence - the team will have a lower velocity, quality, and will consistently overcommit. The Product Owner will be surprised.

10/05/08

Copyright 2003-2008, ADM and MountainGoatSoftware, All Rights Reserved v8.3

Small Modifications



We've made some modification to Scrum to fit our circumstances. Do you have any suggestions?

ScrumBut 2

<We use Scrum, but>

<we need large teams to have all the skills necessary>

<our teams are larger than the recommended 9>

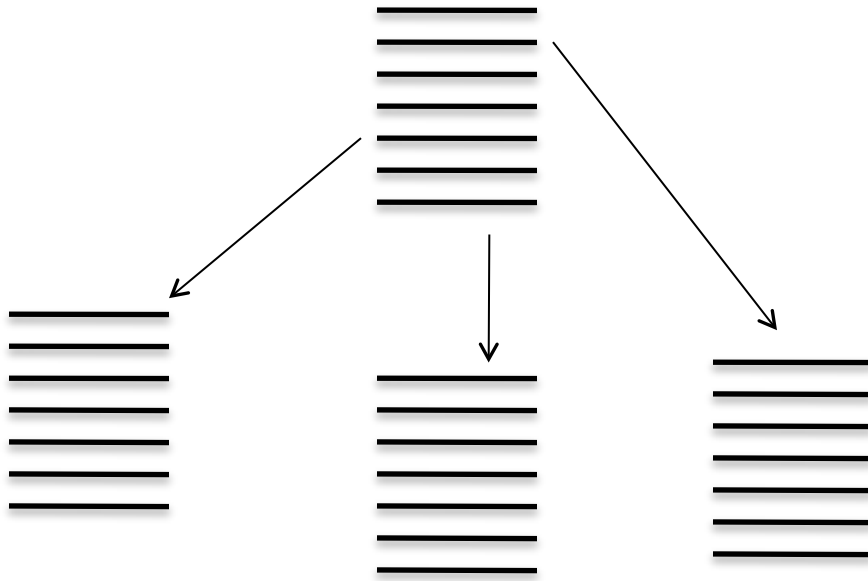
Principle - communication paths overwhelm a team of more than 9 people

Consequence - the team may form subteams (if allowed) or flounder

10/05/08

Copyright 2003-2008, ADM and MountainGoatSoftware, All Rights Reserved v8.3

Team Composition



I parse the initial Product Backlog into sub-backlogs for more teams. However, one area has so many dependencies and requires such specialized knowledge that the team requires *seventeen* team members

What do I do?

ScrumBut 3

<We use Scrum, but>

<we don't have enough people with special, unique skills>

<we have them define their work prior to starting the first Sprint>

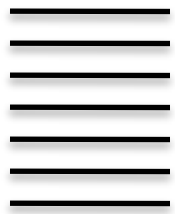
Principle - The product, including architecture and user interface, emerges to fulfill requirements.

Consequence - The architecture and design is fat; quality suffers as teams work to develop the system in the remaining time.

10/05/08

Copyright 2003-2008, ADM and MountainGoatSoftware, All Rights Reserved v8.3

Team Composition



Team 1

Team 2

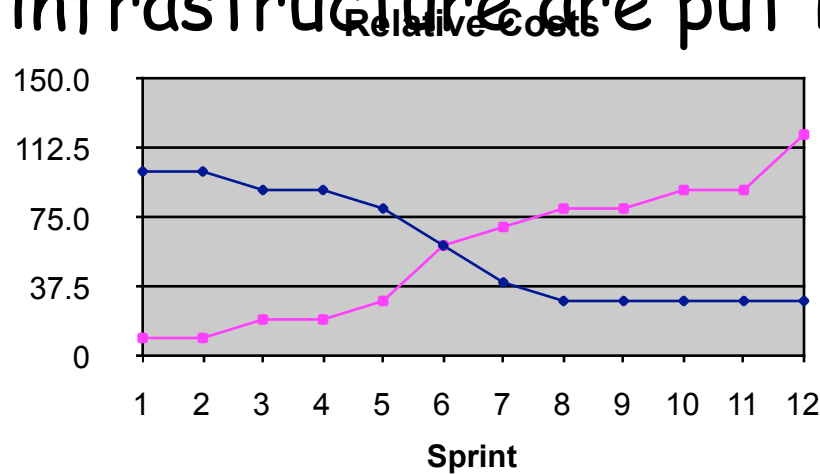
Team 3



I'm SM for three teams. They are all working from the same Product Backlog, have the same Product Owner, and are using the same code base.

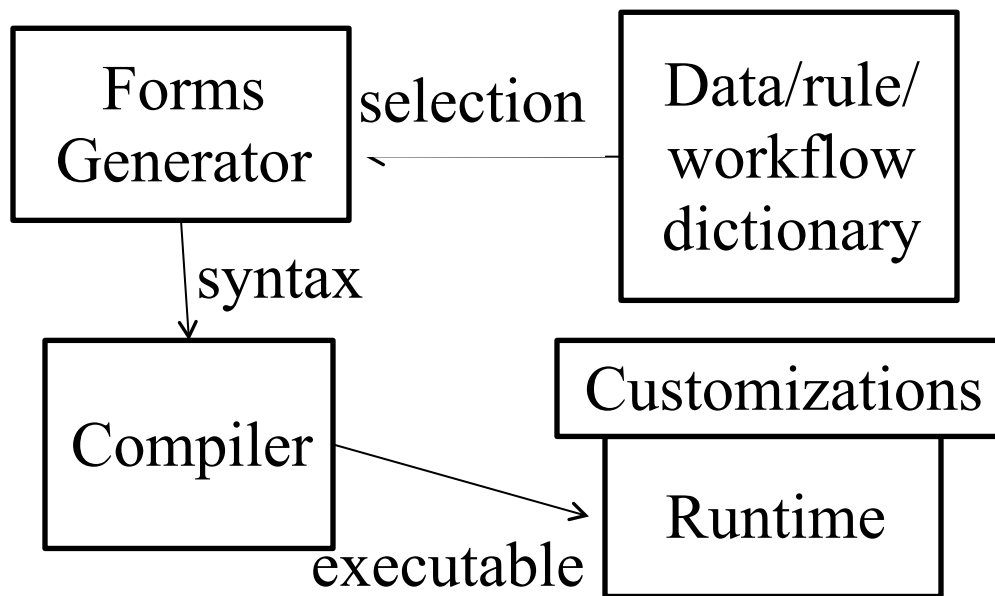
The teams tell me that in the upcoming three Sprints they will all be working in one area of the database; they each need Cindy, the only DBA that knows that subschema well, full time for these three Sprints. What should I do?

Cost allocation changes within the project as architecture and infrastructure are put in place



Exercise : The Art of the Possible

1. Product Backlog Item: Tax preparers need to be able to add customer questions to different forms so that they can serve their customers more specifically.
 - a) Tax preparers need to be able to determine if the demographic of the taxpayers zip code indicates that the number of dependents is excessive on form 1040-a32.



Decompose Product Backlog 1.a so it consists of items of 16 days or less that can be "done" in one monthly Sprint.

Effort Team**PB Item**

72 CTP

Tax Preparer can enter rule zipcode "24760" and demographic "CA" and dependents >10 on form 1040C-a2 so that it causes an exception

16 CTP

Tax Preparer can enter rule zipcode "24760" and demographic "CA" and dependents >10 on form 1040C-a2 into data/rule/workflow engine so that it is entered into the data dictionary

6 CTP

Data selection engine can select form 1040C-a2 from the data/rule/workflow engine so that it also extracts custom rule zipcode "24760" and demographic "CA" and dependents >10

16 CTP

The forms generator can generate syntax for form 1040c-a2 so that rule zipcode "24760" and demographic "CA" and dependents >10 is syntactically correct and recognized by the compiler

16 CTP

Compiler can recognize and act on syntax for form 1050c-a2 with rule zipcode "24760" and demographic "CA" and dependents >10 on form 1040C-a2 so as to generate the appropriate run time code for the runtime engine

12 CTP

The run time engine recognizes and binds the executable so that when form 1040C-a2 is reached the the rule zipcode "24760" and demographic "CA" and dependents >10 can be located at the proper time and executed

6 CTP

Tax Preparer can enter rule zipcode "24760" and demographic "CA" and dependents >10 on form 1040C-a2 so that it causes an exception

All

ScrumBut 4

<We use Scrum, but>

<we don't have a fixed definition of "done" when we commit to "do" a product backlog item>

<so we don't what makes sense and pull the system together at the end.>

Principle - Transparency - the Product Owner must know what he/she is seeing to avoid surprises, minimize risks, and predict..

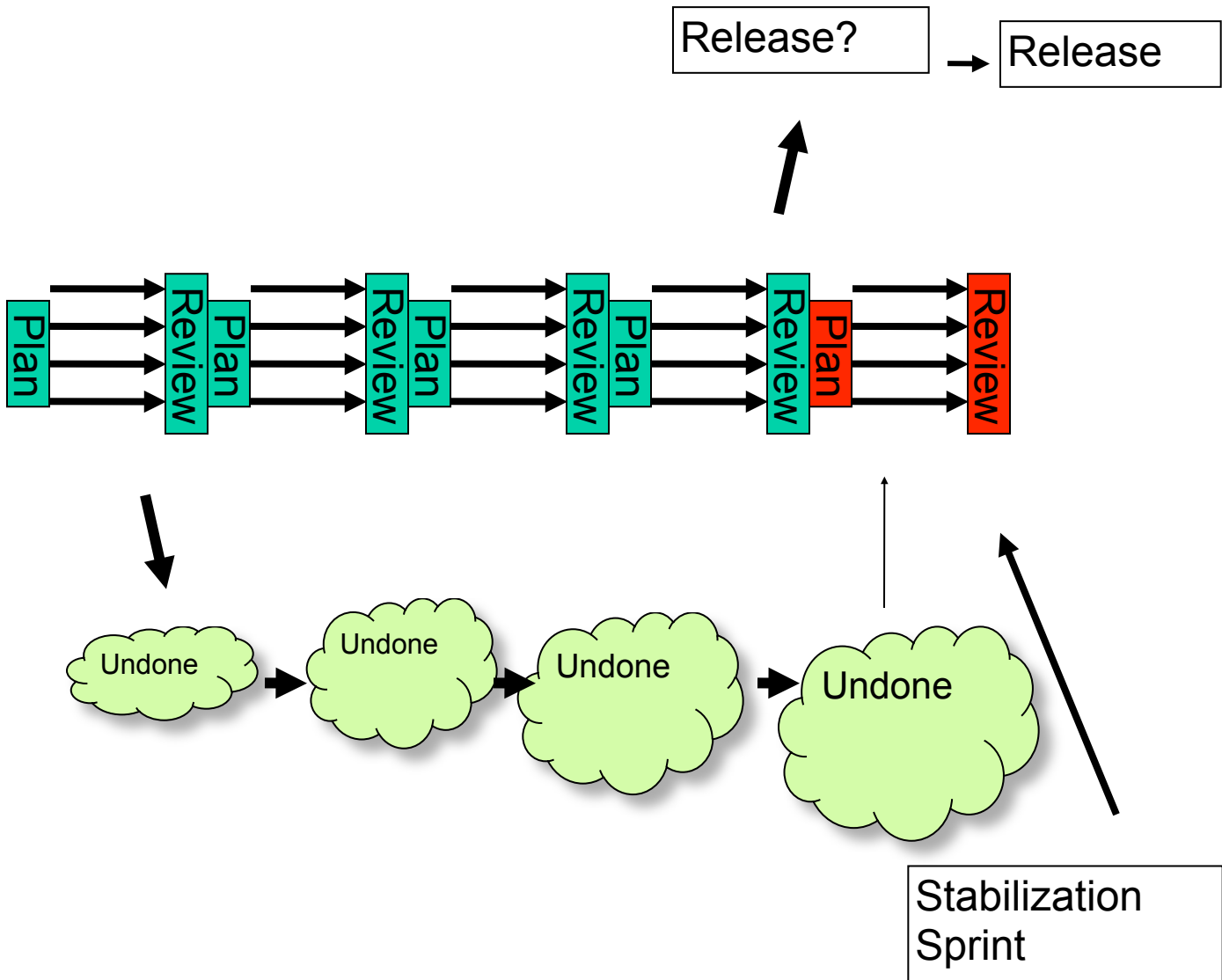
Consequence - II death marches, business as usual.

10/05/08

Copyright 2003-2008, ADM and MountainGoatSoftware, All Rights Reserved v8.3

"Undone" work

35



Create Product Backlog Item
for "Undone Work"

10/05/08

Copyright 2003-2008, ADM and MountainGoatSoftware, All Rights Reserved v8.3

Work Item	Usual	Rec. start	Done
Requirements analysis	25	25	25
Design of architectural components (UI, System, Data	15	15	15
Design review	0	5	5
Design of tests (system, user acceptance, integration)	0	10	10
Design review	0	3	3
Design of documentation	0	2	2
Design Review	0	1	1
Refactoring of existing design	0	0	8
Design of unit tests for new code	0	3	3
Design of unit tests for code to be refactored	0	3	3
Writing new code	10	7	7
Writing refactored code	6	3	3
Code review (or pair programming)	0	4	4
Write functional tests	8	4	4
Write integration tests	0	4	4
Write documentation	4	4	4
Unit test code	0	2	2
Identify and rectify defects	0	2	2
Subsystem/team build	6	2	2
Identify and rectify defects	1	1	1
Unit test for subsystem/team code	0	2	2
Identify and rectify defects	0	2	2
System/integration build	1	1	1
Identify and rectify defects	0	2	2
System, functional tests	1	2	2
Identify and rectify defects	1	2	4
Integration tests	0	0	2
Identify and rectify defects	0	0	5
Performance tests	0	0	1
Identify and rectify defects	0	0	2
Security tests	0	0	1
Identify and rectify defects	0	0	2
Regression test	0	2	2
Identify and rectify defects	0	8	8
Documentation test	0	1	2
Identify and rectify defects	0	1	1
Total work expended requirement	78	118	148
Work remaining per requirement	65	30	0

36

"Done" is
not
defined.



1. No stable velocity from which to estimate;
2. Inaccurate product backlog burndown;
3. Product Owner doesn't know progress or status;
4. The Product Backlog probably isn't in good shape;
5. Team doesn't know how much to select in Sprint Planning meeting; and,
6. Product Owner doesn't know what is being inspected at Sprint Review.

ScrumBut 5

<We use Scrum, but>

<we don't need the whole Sprint Planning meeting because we know what we are going to do>

<so we don't waste time.>

Principle - Complex product development has more unknowns than knowns. The team has to lay out an initial plan that it can inspect and adapt to.

Consequence - II death marches, poor quality, overcommitment, nothing "done" at Sprint Review.

10/05/08

Copyright 2003-2008, ADM and MountainGoatSoftware, All Rights Reserved v8.3

The Team doesn't design how to meet its commitment in the Sprint Planning Meeting



1. Team probably doesn't know that the sum of the tasks will deliver its commitment;
2. Team cannot self-manage;
3. Cross-functionality probably isn't happening.

ScrumBut 6

<We use Scrum, but>

<we don't have time/need for a
Daily Scrum meeting>

<so we only hold them weekly/as
needed>

Principle - a self managing team
inspects its progress toward meeting
its Sprint commitments and adapts
to optimize its chances

Consequence - the team will have a
lower velocity, quality, and will
consistently overcommit.

10/05/08

Copyright 2003-2008, ADM and MountainGoatSoftware, All
Rights Reserved v8.3

The Daily Scrum is a status meeting



1. Team isn't self-managing;
2. Team has no idea where it is toward delivering its commitment;
3. Team will probably not deliver everything that it has committed to; and,
4. If "done" is defined, it may be adulterated in those things demonstrated at the Sprint Review.

Scrum Is Simple

.....

Fixing What It Uncovers

Is

Very,

Very,

Very,

Hard



www.controlchaos.com/scrumbut.pdf

10/17/08

Copyright 2003-2007, ADM, All Rights Reserved v1.1