

Introduction to Agile Methods and Practices

Hubert Smits

July 23, 2006

Purpose of this Session

Let's work as a group, first by thinking about what we want to learn this afternoon. Then we'll talk about the topics – the most important ones first. I can guide you, try to explain subjects. And you can make the afternoon a true success. By asking questions, bringing experience (good and bad) and by being open minded. This afternoon could be a first step in changing your life. Forever.

Enjoy.

Me

I'm Hubert Smits, I'm an Agile Coach for Rally Software Development. After spending some 20 years in managing large software development projects in Europe I saw the light, joined the Agile Alliance and became an agilist. First an agile project manager (ScrumMaster), then a trainer and coach. I now run Agile Implementations in the US, Europe, the Middle East and India. I've done some lecturing on Agile Software Development at Glasgow University. My main community is still Scrum, I am privileged to be able to train ScrumMasters, and I enjoy working with some of the thought leaders in this area.

You

Do you know about agile? Maybe practiced it? You love it?
Like it? Hate it? Are you at the wrong conference?

An Agile Workshop

Agile is simple (not easy). Agile is about *people*, about doing the *important things first*, and about *taking small steps*, learning and *improving* as you go along.

Let's apply the main rules to this session. You are the people, you tell me what is important. Then we'll talk a few minutes about each topic and at regular points we'll check in to see if this is working for you.

Topics

To get you going, here are a few topics we could talk about.
Let them not limit you, let them start you.

Scrum

Large Team

Practices

Small Team

Principles

How it all started

Off-shoring

Do's and Don'ts

Scaling

People - Roles

Impact

Does it work?

Lean

Extreme Programming

The Agile Manifesto

“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

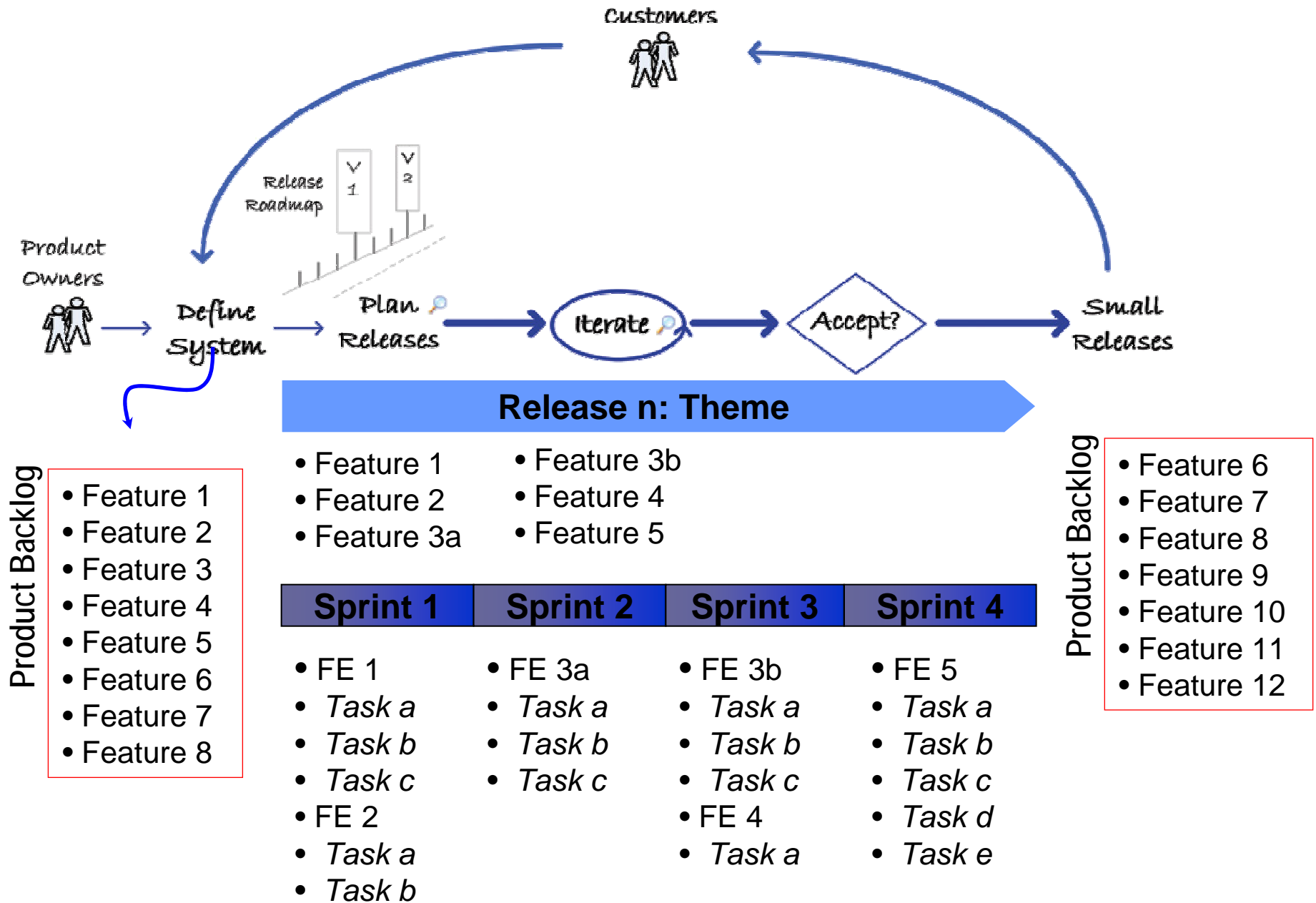
That is, while there is value in the items on the right,
we value the items on the left more.”

Defined vs. Empirical

“It is typical to adopt the defined (theoretical) modeling approach when the underlying mechanisms by which a process operates are reasonably well understood.

When the process is too complicated for the defined approach, the empirical approach is the appropriate choice.”

An Agile Process Diagram



Agile Roots

Hiroataka Takeuchi & Ikojuri Nonaka

- The New New Product Development Game
- Harvard Business Review – Jan/Feb 1986

Family Album



Takeuchi

Nonaka



Poppendiecks



Ambler



Schwaber

Sutherland



Beck



Cunningham



Jeffries

The Agile Mythology

There is often stories about lack of *discipline*: “Agile lets my Engineering Teams do whatever they want” and “Quality of the product will fall off”. Or about lack of *visibility*: “I have no view into what is happening” and “I can’t predict what I will get, or when”. And we can’t do this here (lack of *applicability*): “Agile is just for software geeks” or “Agile is just for small teams”.

And most often: “Agile is easy”!

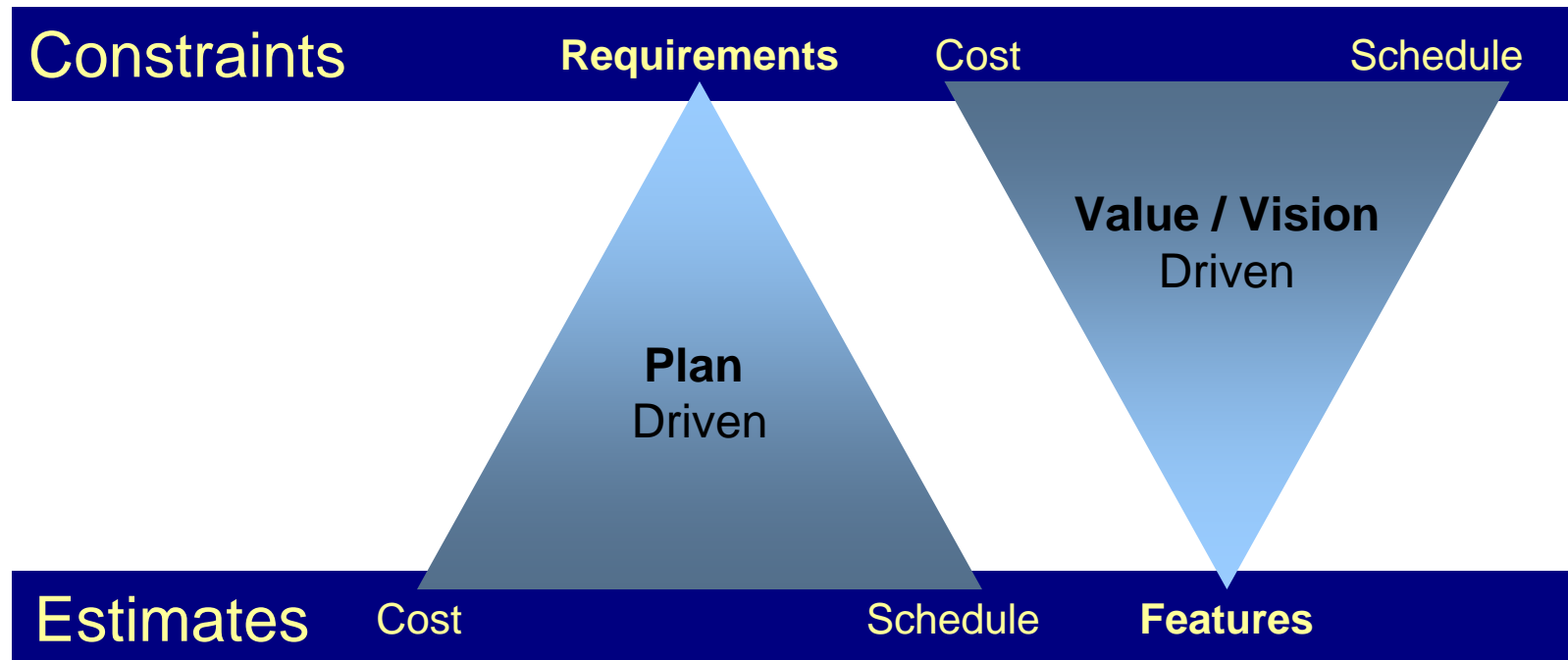
Project Drivers

Waterfall

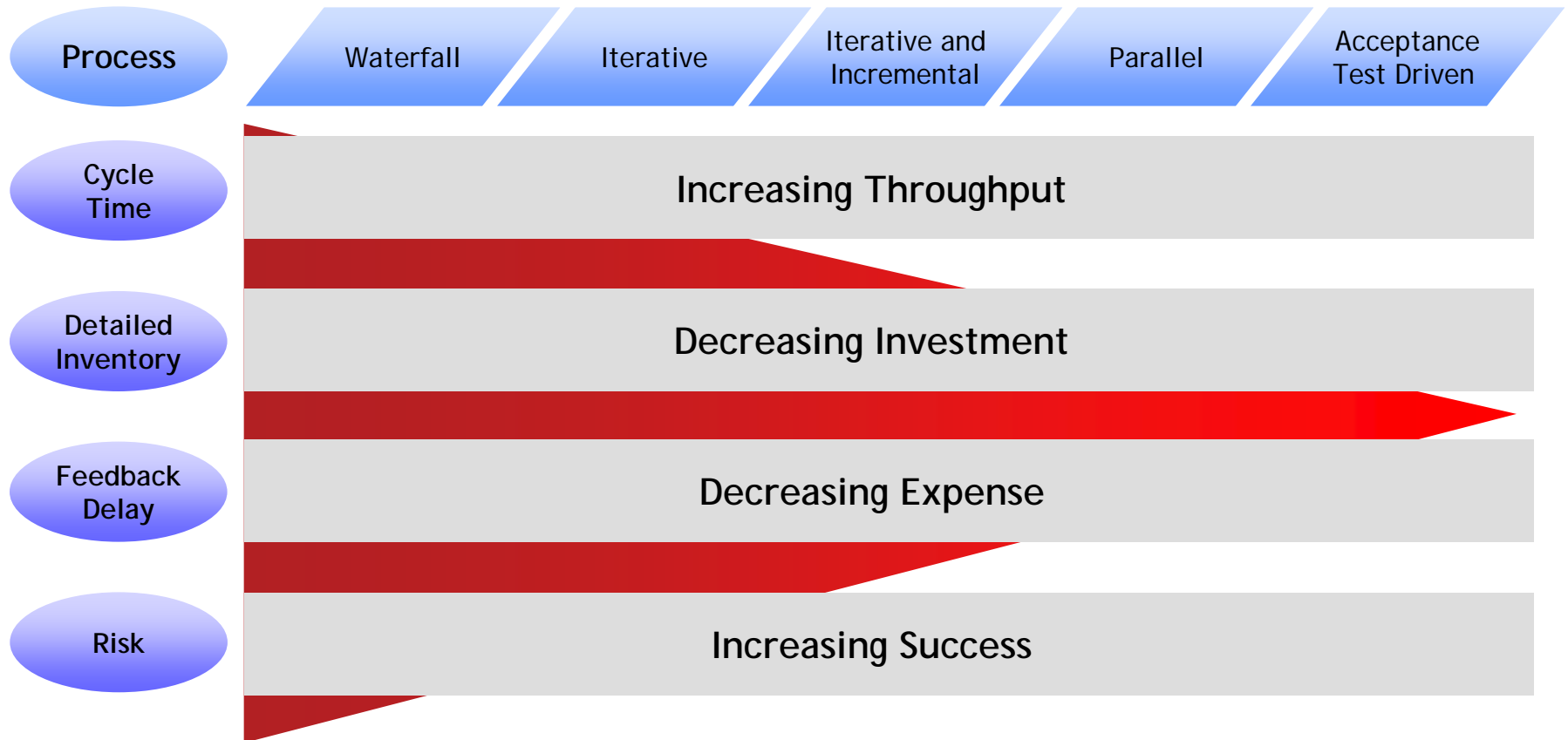
*The Plan creates
cost/schedule estimates*

Agile

*The Vision creates
feature estimates*



Success Criteria



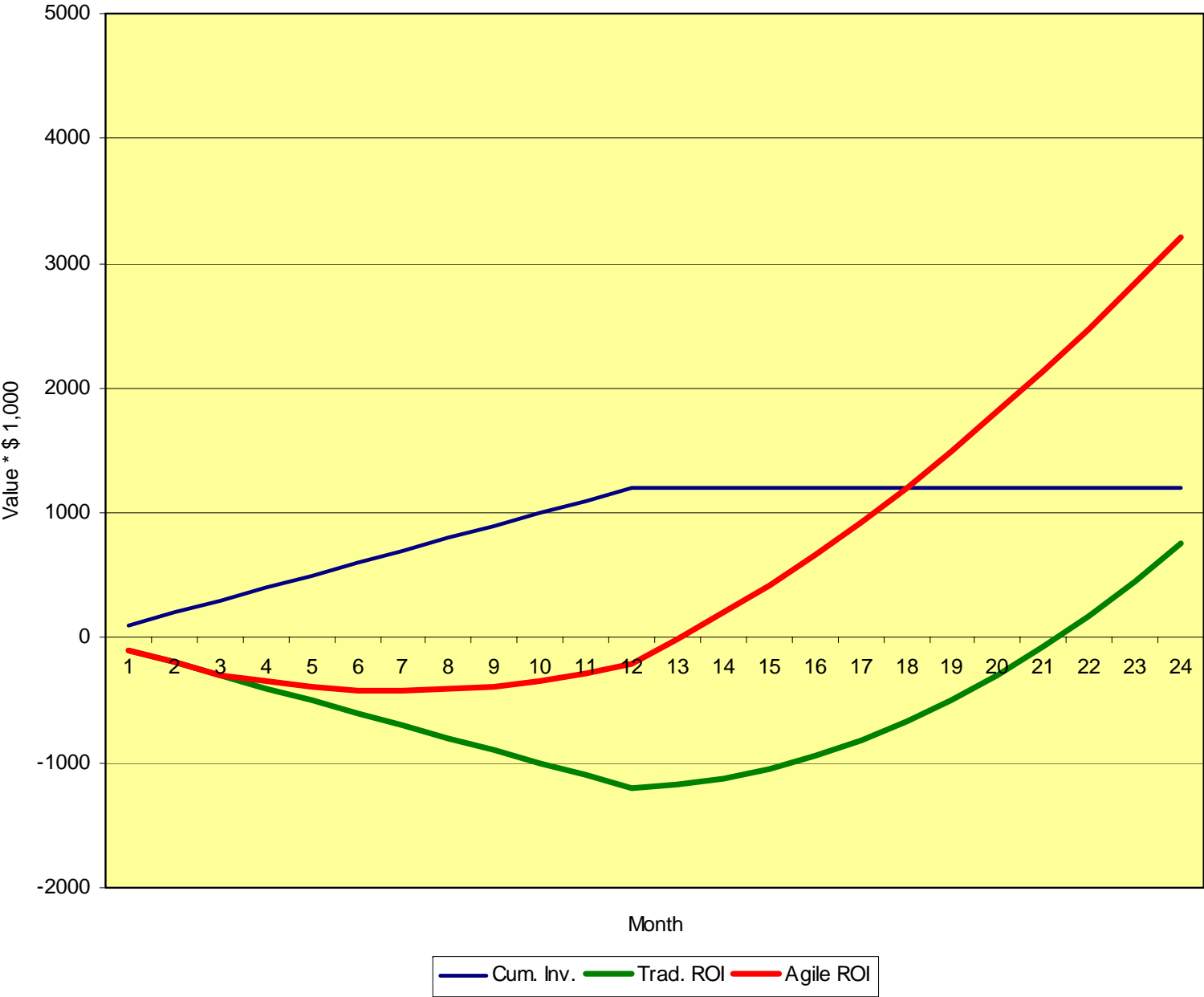
Measures of Success

Process	Waterfall Development	Iterative Development	Iterative and Incremental Development	Parallel Development	Acceptance Test Driven Development
Cycle Time & Feedback	Year+	Quarter+	Month+	Week+	1 - 2 Weeks
Detailed Inventory	Whole Project	Phase	Iteration	Increment	Feature
Risk for Team of 10	\$ 1,200,00	\$ 450,000	\$ 200,000	\$ 100,000	\$ 50,000
Tool Support	Role Specific Tools	Batch Integrations	Real-time, collaborative project and work product management		

Measures of Success



Comparison – Traditional versus Agile



Quantitative & Qualitative Results

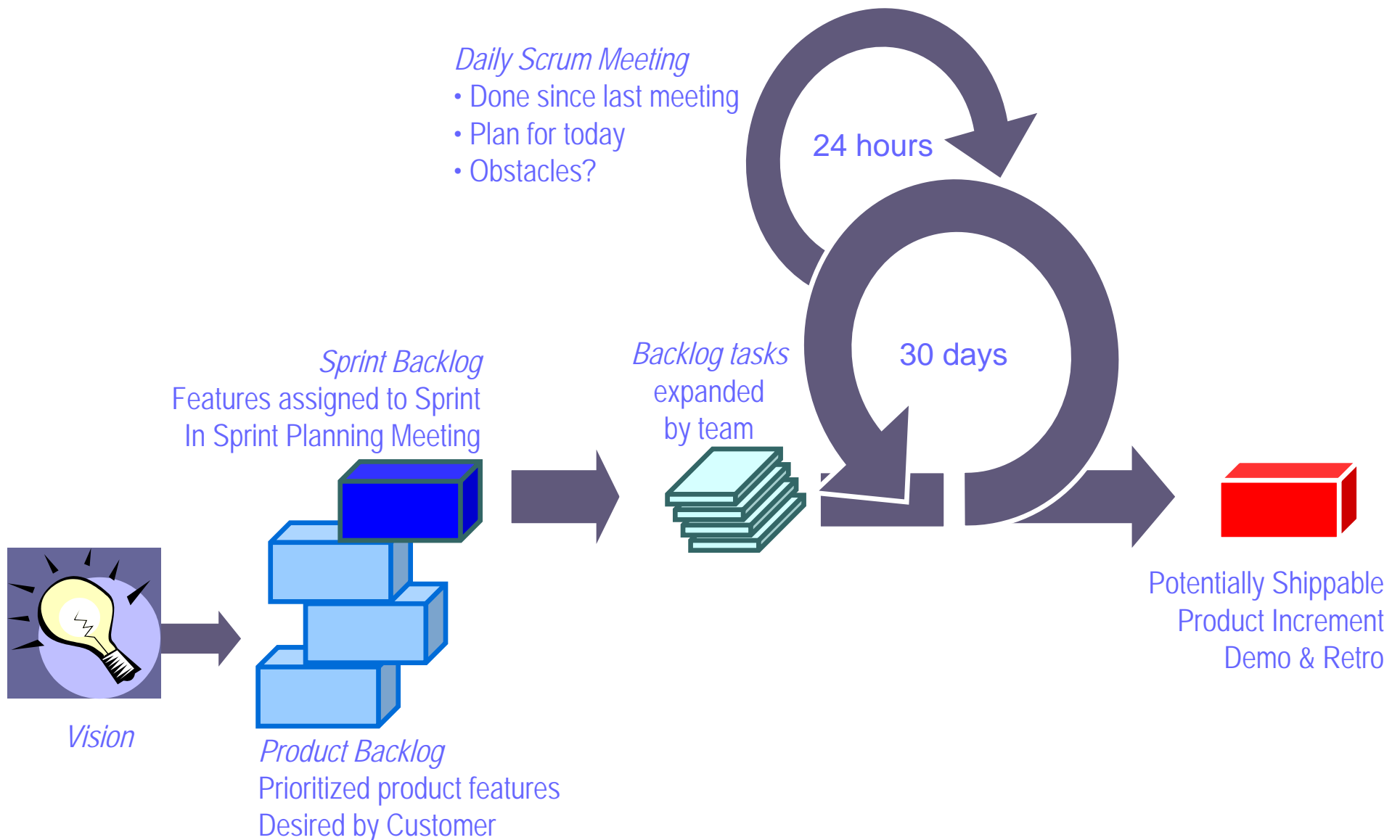
Forrester Total Economic Impact Studies (1)

- 5 Companies piloting Agile methods
- 3 yr, Risk-adjusted ROI of 23% – 66%

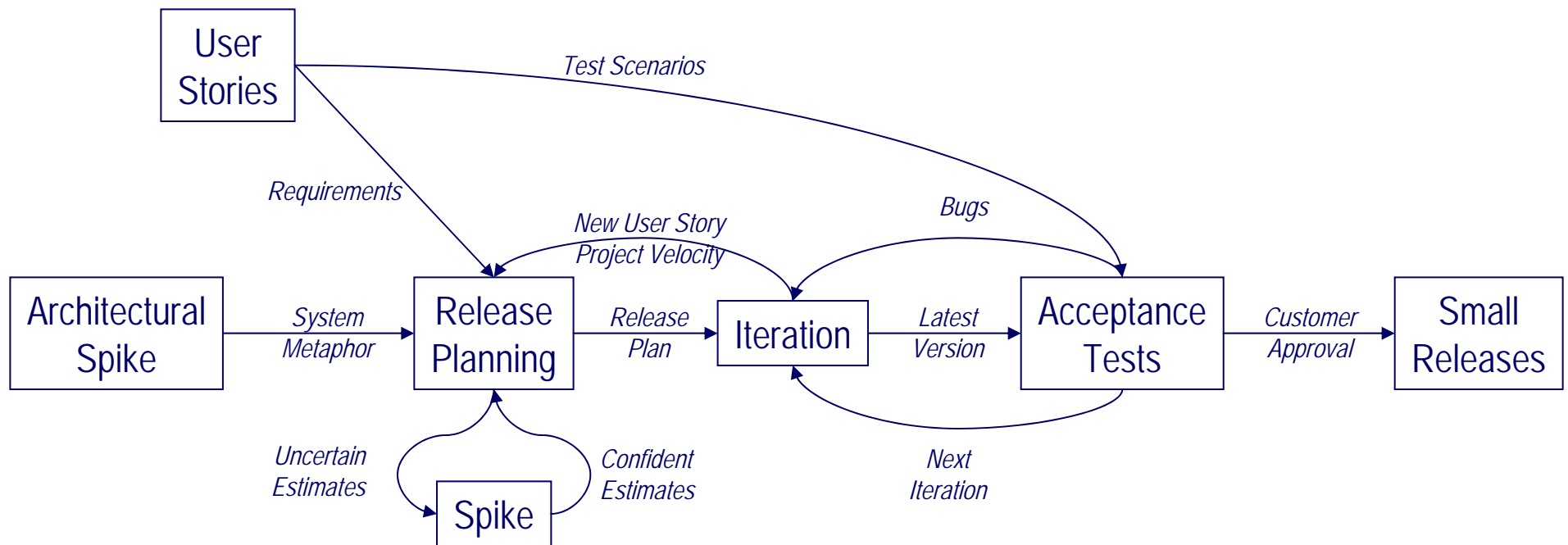
Agile Methodologies Survey (2) , 131 respondents:

- 49% stated that costs were reduced or significantly reduced, (46% stated that costs were unchanged)
- 93% stated that productivity was better / significantly better
- 88% stated that quality was better / significantly better
- 83% stated that business satisfaction was better or significantly better

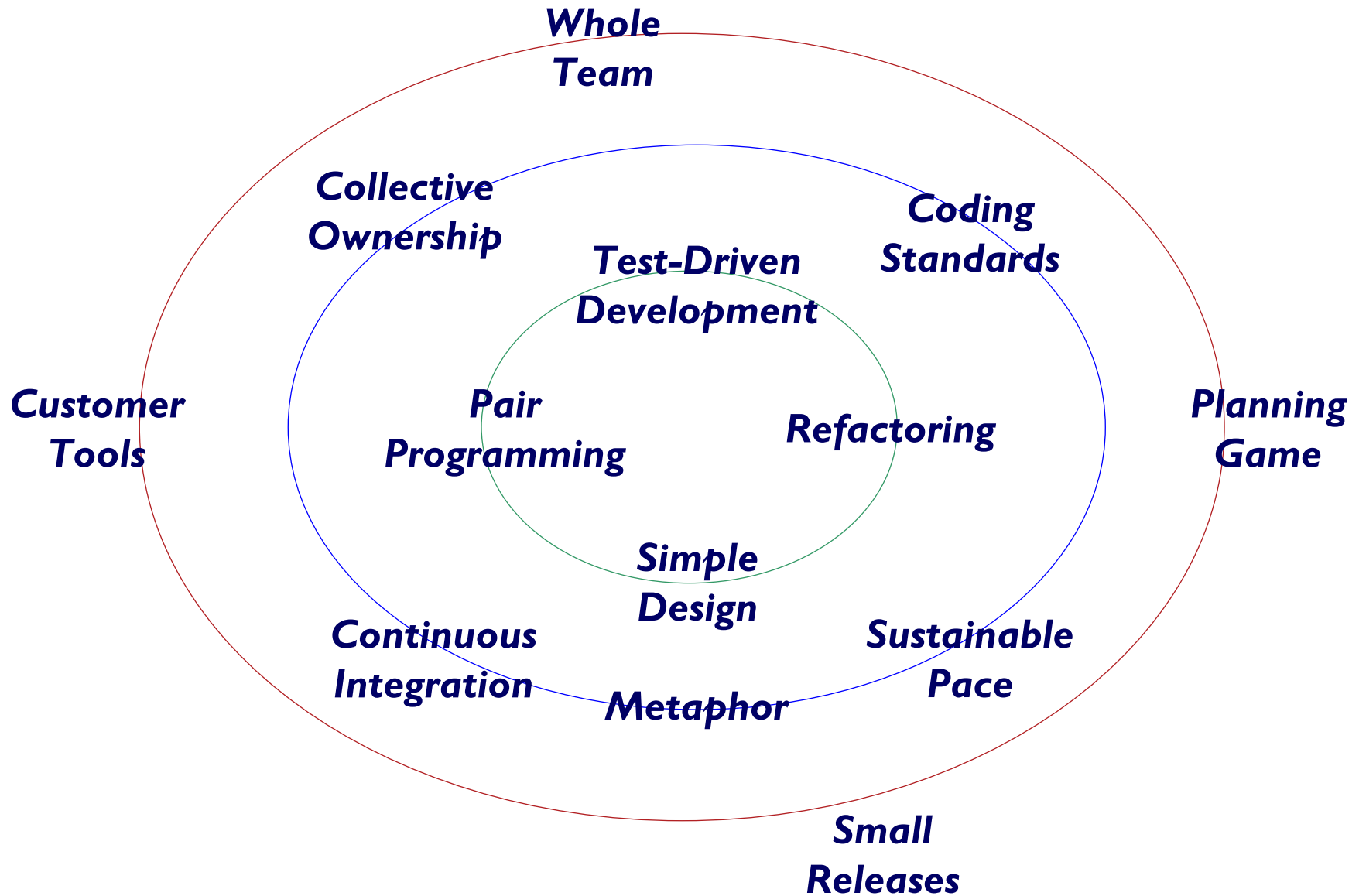
Project Management Method Scrum



An Engineering Framework - XP



XP Practices



Common Practices*

What happens in teams (after going through a couple of iterations) focuses often on engineering practices like *Continuous Integration* through *Build Automation* (early surprises), and *Automated Deployment* (give every result to the product owner to play with).

Team practices are typical Simple Design or “do the simplest thing that could possibly work.” Adding complexities is followed by *Refactoring* or “first make it work, then make it pretty.” The *Collective Ownership* of the system is achieved by Pair Programming – and vice versa!

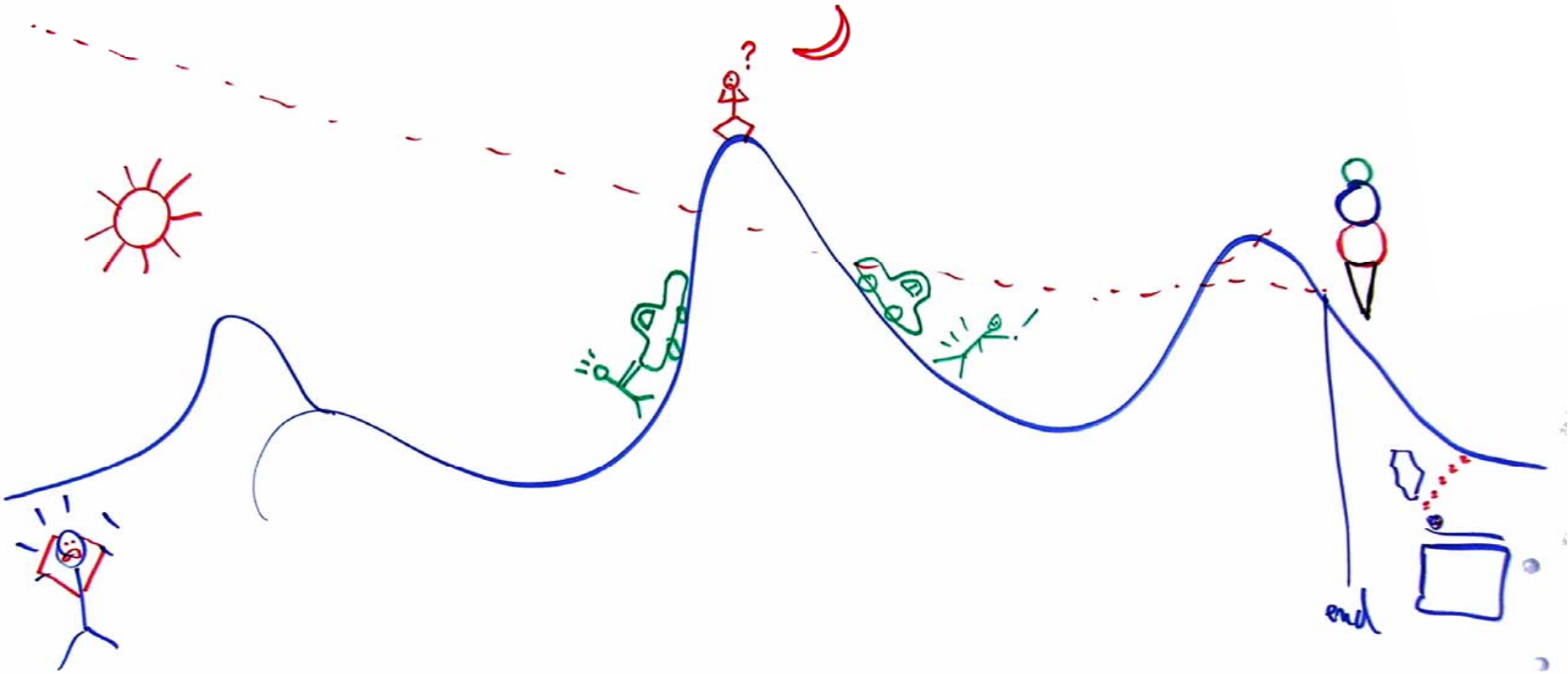
If you grow the agile initiative, the create *Feature Teams* – don’t separate functions (like the test-team in Bangalore.)

* Peter Schub – *Integrating Agile Development in the Real World*

Inspect and Adapt



Sample Retrospective



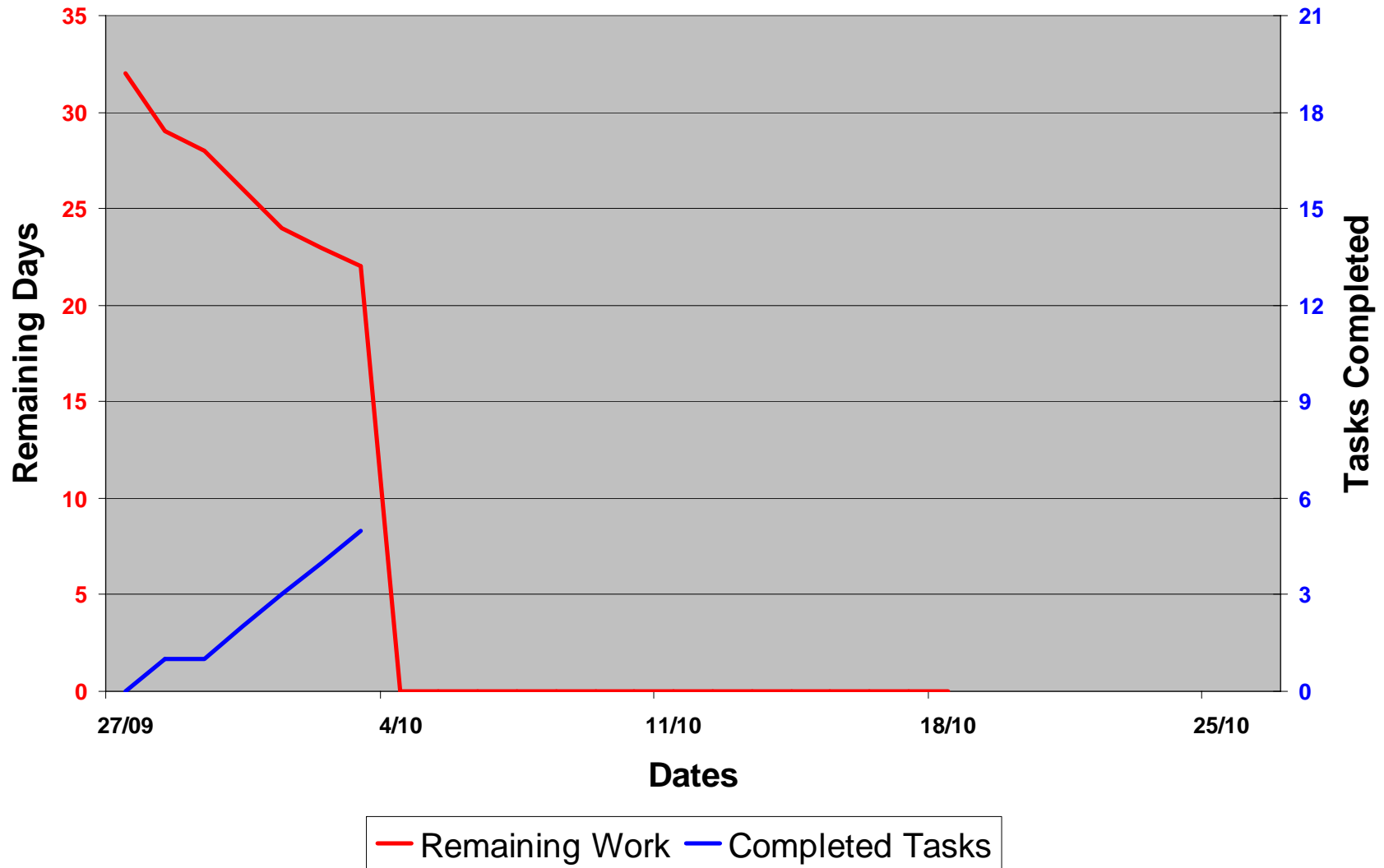
Retrospective

Lessons for today

- * Better planning preparation
- * Better individual capacity planning.
- * Risk mgt.
- * Don't commit to unknowns,
commit to the investigation
(commit to commit)
- * Milestone planning - builds, release notes etc.
- * Consider impact of Reviews & Short tasks
- * Impact of templates.
- * Have team productive early

Burndown Chart

Remaining work (days) for project X phase Y



Product Owner / Product Manager

Defines a Product Development Roadmap, indicates the focus (theme) and timing of the next few releases. She prioritizes stories in the Product Queue (Backlog). Works intensely with the team, presenting her vision to the development team, and maintaining enough detail of each story in anticipation of the next level of planning. And: she negotiates about priorities, about techie needs vs. business needs.

The Delivery Team

The delivery team delivers the goods, every iteration, in a constant rhythm, at a pace that they can maintain for years. This is a big responsibility, so it comes with privileges. For example, they own the estimates, and they can self-organize. They decide as a team on how to deliver the goods that they committed to deliver in this iteration. They can swap roles, they may well have to do this to be able to all work at a sustainable pace. And it is a delivery *team*, everybody who has a role in producing a result that the product owner can accept is in the team. Architects, testers, QA people, developers and business people. Since the only measure of progress is working software, this is how they report progress: a daily estimate of the remaining work to deliver the goods. And they tell their immediately leader what stops them from delivering the goods on time.

Project Manager / Scrum Master

She's no longer responsible for the success of the project! Her main concerns are to safeguard the process, keep the bad folks (aka managers) away from the team. She removes roadblocks and acts as a sheepdog for the team. She protects the team (management) and eliminates distractions (management). She'll work with the team to foster team communications, giving them the tools they need to be successful. Or: *she maintains unrelenting dedication to team's success in meeting its commitments.*

For short: she is a Servant Leader.

Other Team Advisors

Pigs and Chickens — a short story by Ken Schwaber

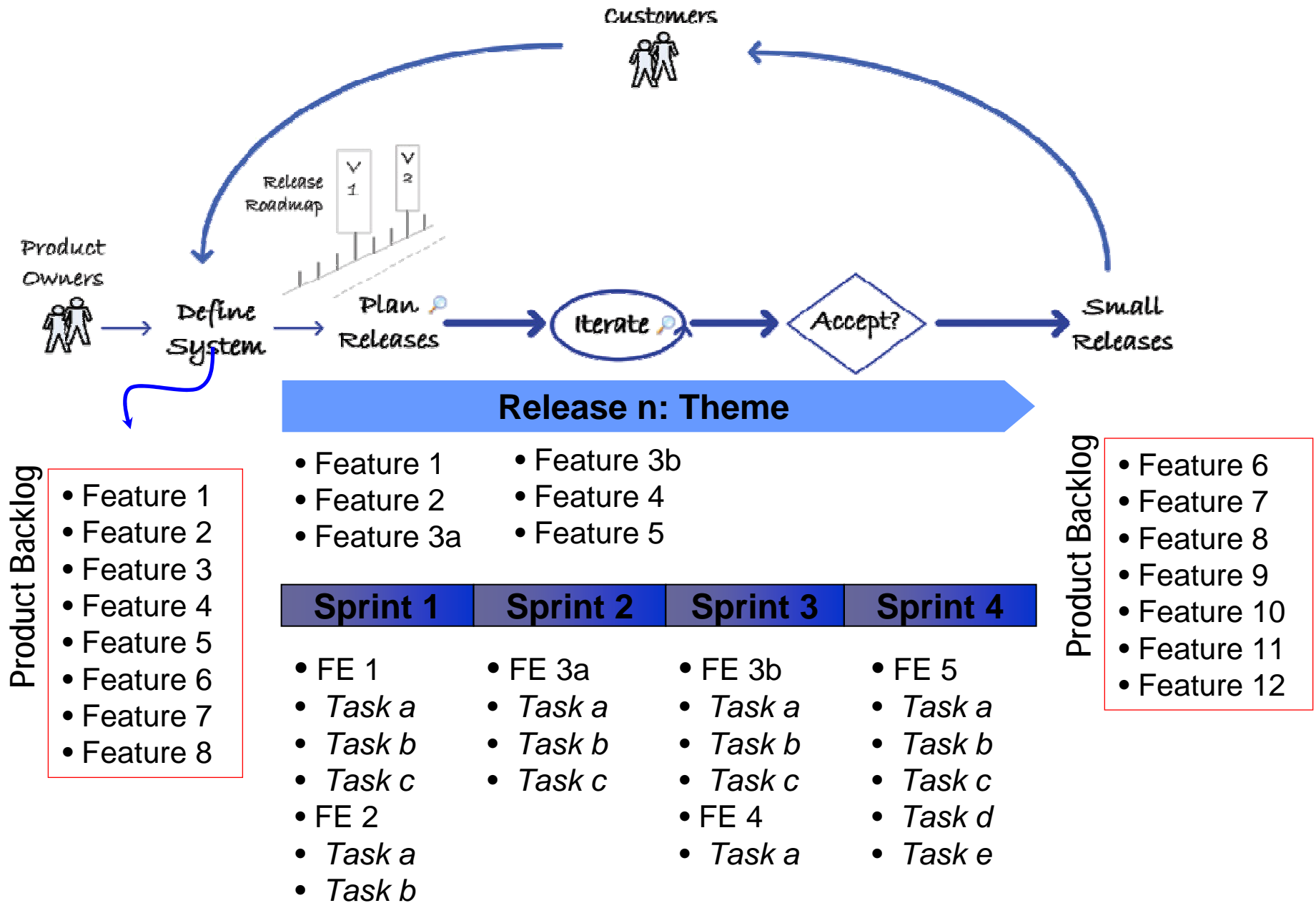
Agile Set in Motion

Just do it!

Choose a method you like, Scrum is a simple and easy start. Choose a pilot project, and a pilot team (maybe two). Make a (short) list of the most important features you need. The list doesn't have to be complete, as long as it is prioritized – you can update your list as often as you like. And now the agile train starts its perpetual journey.

The developers plan the features – together with you, they will need some explanations on what you mean. They pick just enough work for the first iteration, and they deliver the features. At the end of the iteration all inspect the results, and the process. You agree on improvements for both, and the cycle starts again. Simple. Not easy.

An Agile Process Diagram



Iterations vs. Releases

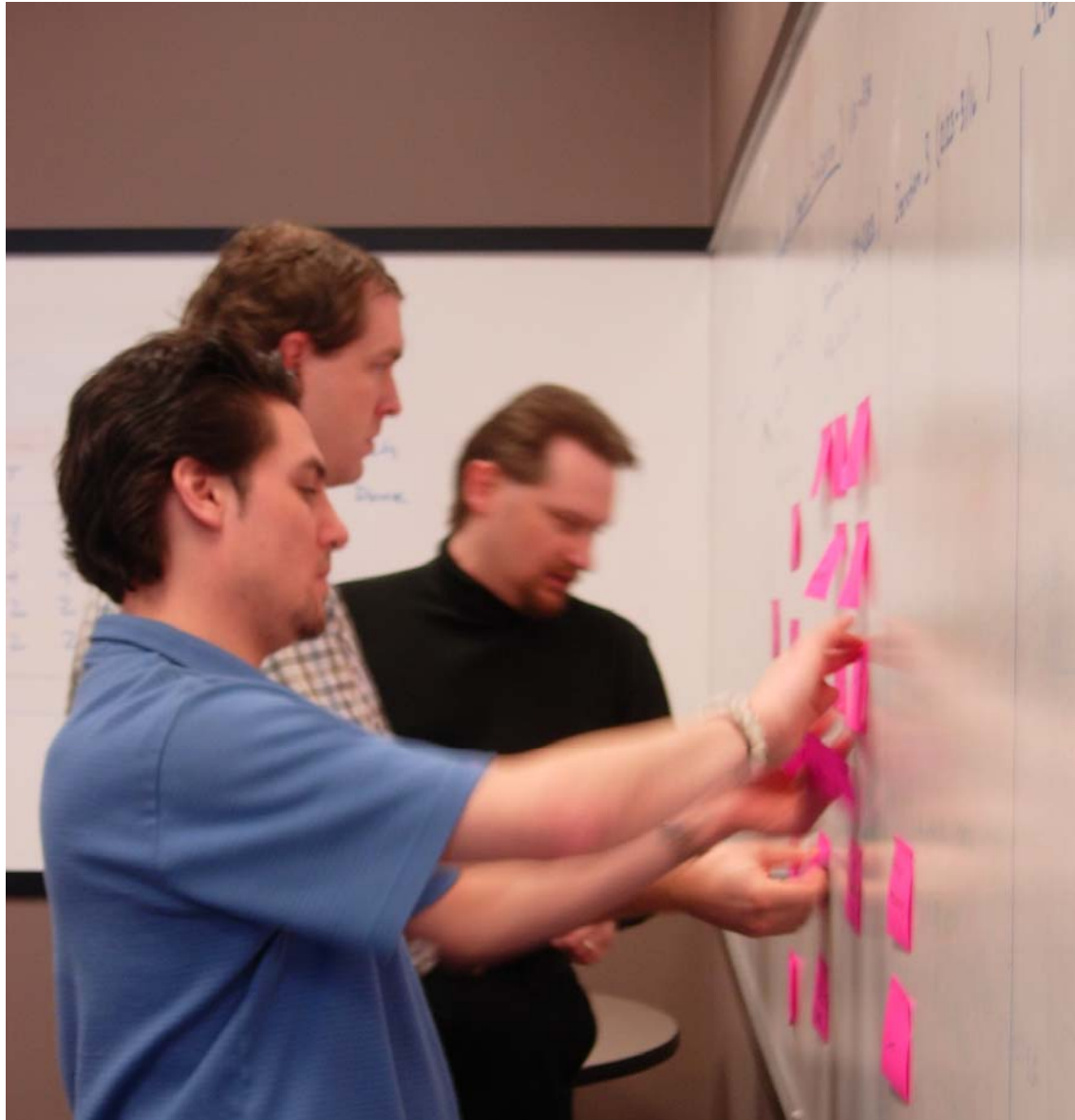


Sample Planning Agenda

Agenda

1. Introductions, Agenda, Goals
2. Product Vision & Metaphor
3. Timeboxes for Release & Sprints
4. Development status - prev. Sprints
5. Teams - capacity / T-Shirt sizing
6. "Done" revisited
7. Feature mix:
 - New features, rework, bugs
 - Generic support (build etc.)
 - Tool testing
 - Next Sprint prep^s
8. Breakout - Move features in Sprints
9. Together - Review
 - Workload
 - Dependencies
10. Risks, Issues, Mitigations
11. Commit
12. Retrospective

Sample Planning Session



Commitment



Planning Room



Closing

Next Steps

Given what you have learned here today,
what will you do when you're back in the office?

Thank you!

hubert@rallydev.com

Exercise – Command and Control

We're working in pairs, look at your neighbor and introduce yourself. For the exercise, one of you is to be the boss, the other the worker. You pick.

The exercise is as follows: the boss tells the worker how to make 60 steps through the room within a minute. He can do this with the simple commands – Go, Stop, Right, Left, Faster, Slower.

And the worker does what the boss tells him to do. He also counts the steps.

Don't hurt yourself! Nor others!

Exercise – Self Organizing

Ok, remember what happened, and let's make a small change. With the same teams as before, the workers can now decide how to make the 60 steps, and the bosses remove any obstacles the worker encounters. Do leave the walls in place! This time you have 30 seconds.

Exercise: The Art of the Possible

I need two assistants, don't take this too serious, I won't make a fool of you (you might). Assume that after this session you would like to hang out together, the whole group!

One of the volunteers brings up the idea: "Let's keep this team spirit alive and ..."

The second volunteer responds, starting with "Yes, but ..."

Continue the conversation, volunteer 1 brings on the ideas for the night out, volunteer two starts all the responses with "Yes, but ...".