
think·box
INTRODUCTION TO AGILE AND SCRUM
Presented By
Simon Baker
Scrum Alliance
Certified ScrumMaster Practicing

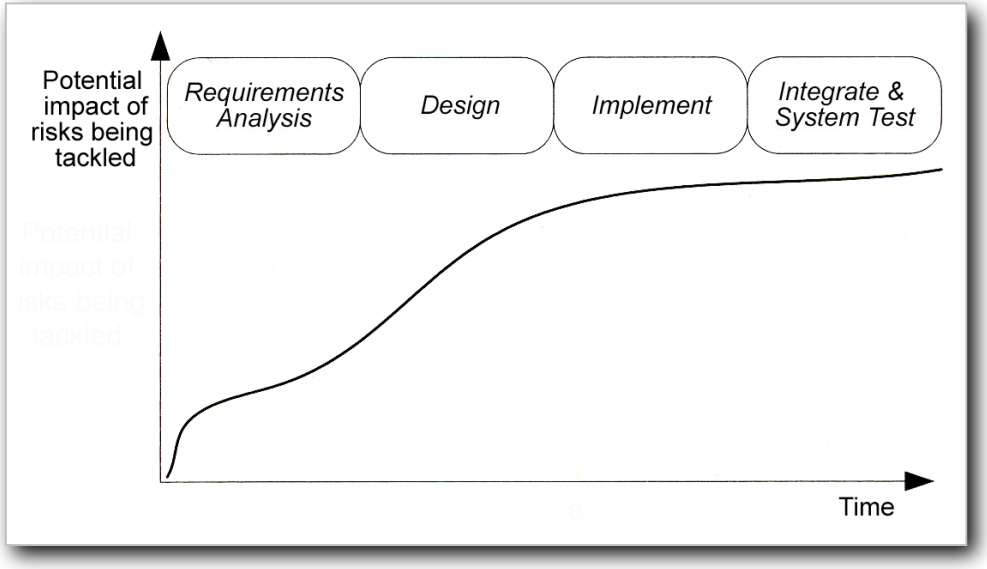
AGENDA

- Waterfall [2 slides]
- Agile [21 slides]
- Basic planning concepts [5 slides]
- Scrum [23 slides]
- Scaling Scrum [8 slides]
- Next Steps [2 slides]
- Closing remarks [1 slide]

Copyright © 2003-2006 think-box Limited

WATERFALL APPROACH

- Defined, predictive process
  - Sequence of distinct phases
  - Plans everything up-front



- Fail-late lifecycle
  - Most of the risk and difficult work is pushed toward end of the project

Copyright © 2003-2006 think-box Limited

PROBLEMS WITH WATERFALL

- Whole project planned up-front
- Doesn’t handle change very well
- Requirements specifications are an abstraction and can be interpreted differently
- Business engagement is high at the start of the project but then tapers off
- Insufficient testing during development
- Late integration
- QA is trailer-hitched, so quality isn’t baked in and testing gets crunched at the end
- Progress measured by task % complete
- Often don’t know until it’s too late

Copyright © 2003-2006 think-box Limited

AGENDA

- ☑ Waterfall [2 slides]
- Agile [21 slides]
- Basic planning concepts [5 slides]
- Scrum [23 slides]
- Scaling Scrum [8 slides]
- Next Steps [2 slides]
- Closing remarks [1 slide]

Copyright © 2003-2006 think-box Limited

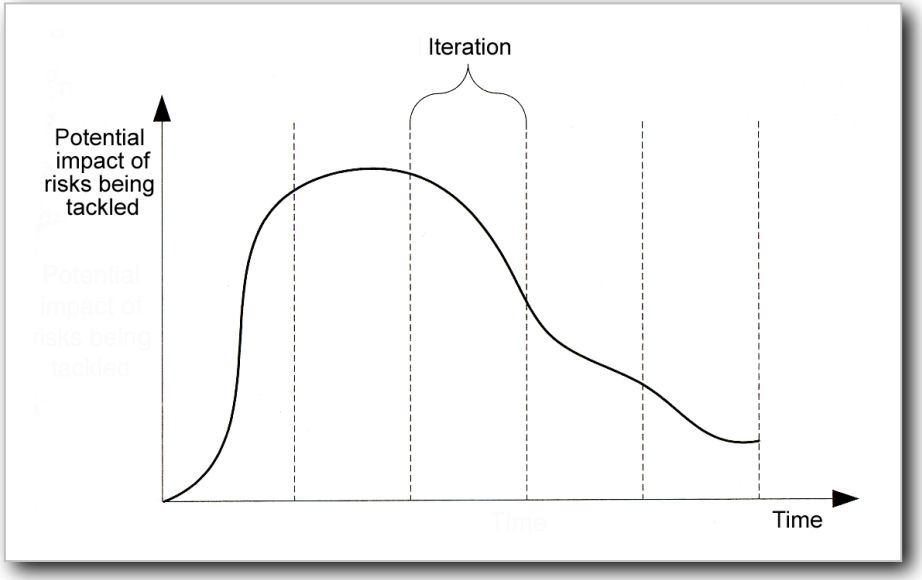
REASONS TO USE AGILE METHODS

- Incremental approach breaks complex projects down into simpler mini-projects
- Accommodates change easily
- Improves ROI through frequent and regular delivery of value to the business
- Increased business involvement and satisfaction
- Increased visibility (progress, obstacles, risks, etc)
- Lower development risk, higher quality, less defects
- Shorter cycles produce working software and incremental product quickly
- Progress measured by running tested software
- Early and regular process improvement driven by frequent inspection

Copyright © 2003-2006 think-box Limited

AGILE APPROACH

- Adaptive, empirical process
  - Small repeating cycles
  - Short-term planning with constant feedback, inspection and adaptation



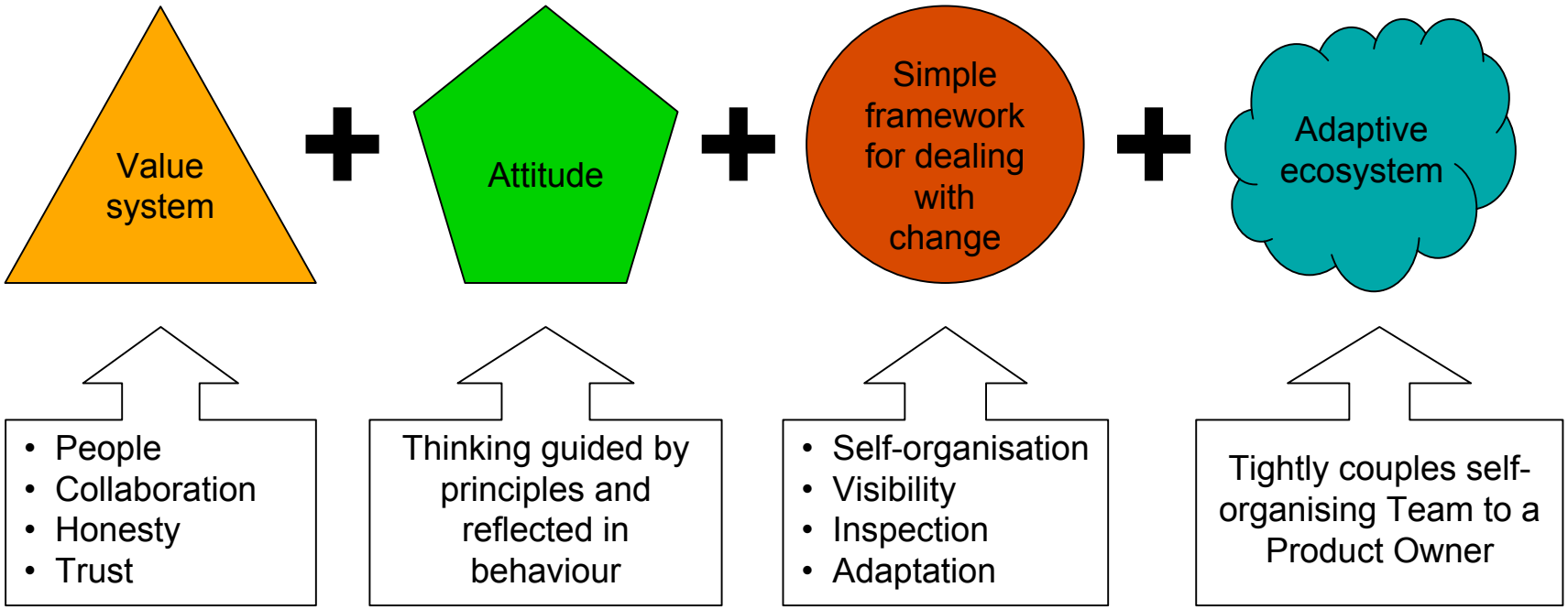
- Fail-early lifecycle
  - Iterative development brings most of the risk and difficult work forward to early part of project

Copyright © 2003-2006 think-box Limited

WHAT IS AGILE?

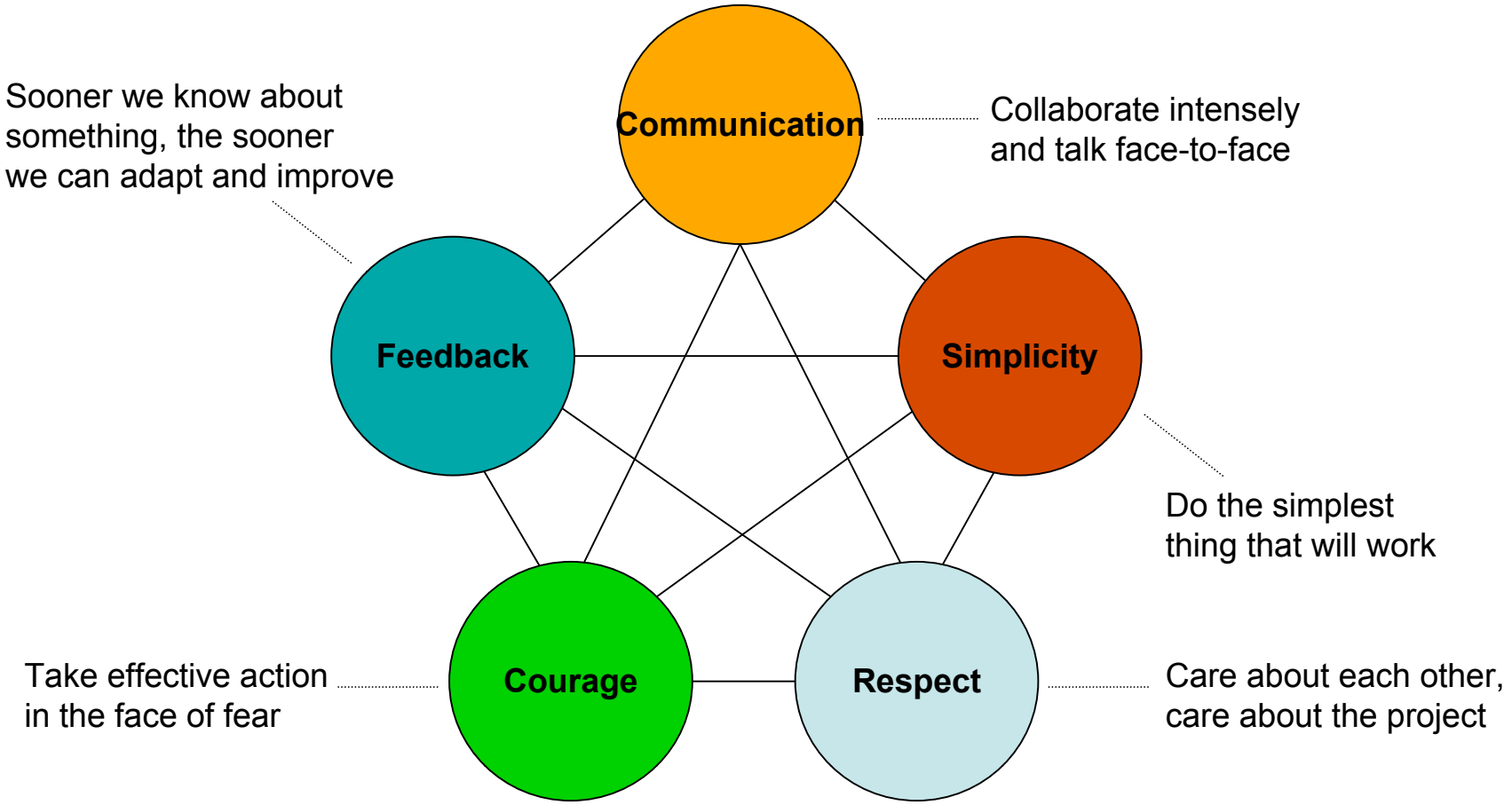
Agile is about being open about what we're capable of doing, and then doing it

– Kent Beck



Copyright © 2003-2006 think-box Limited

VALUES



Copyright © 2003-2006 think-box Limited

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.

Copyright © 2003-2006 think-box Limited

CREATE A SHARED VISION

- **Unites** us
- Guides our decisions
- Post publicly
- Charter
  - Measures of success
  - Project assumptions
  - Project constraints

Copyright © 2003-2006 think-box Limited

SATISFY THE PRODUCT OWNER

- Highest priority
- **Focus On Purpose, Not Process**
  - Concentrate on building product rather than serving process
- Build product that delivers business value early and continuously
- Welcome changing requirements
  - Harness change for a competitive advantage

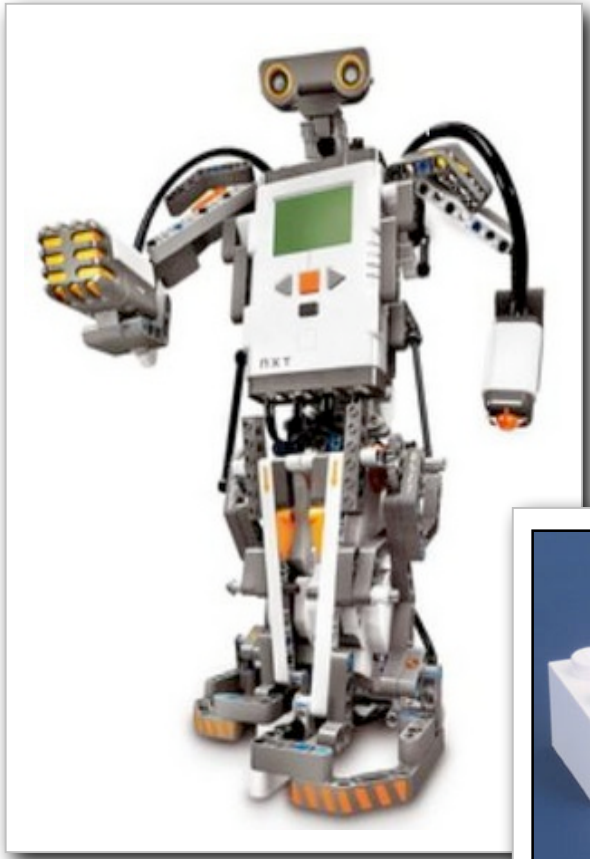
Copyright © 2003-2006 think-box Limited

STEADY FLOW

- Arrange work in a prioritised queue that delivers a steady **flow** of running tested software (weekly/monthly/quarterly)
- Product Owner should **pull** business value from the Team
- Use energised work to maintain a **constant pace** indefinitely

Copyright © 2003-2006 think-box Limited

THINK BIG. START SMALL



- Get something small and hard-hitting out there early
- Collect feedback from real users
- Build on it quickly with **regular incremental releases**

Copyright © 2003-2006 think-box Limited

WORK FROM THE OUTSIDE, IN

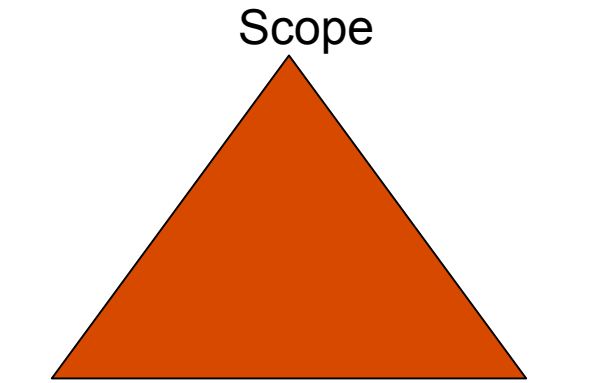
- Let the **users drive your efforts**
- See everything from the user's point of view
- Understand value from the user's perspective
- Ask the users what they want next

Copyright © 2003-2006 think-box Limited

FIX TIME AND BUDGET. VARY SCOPE

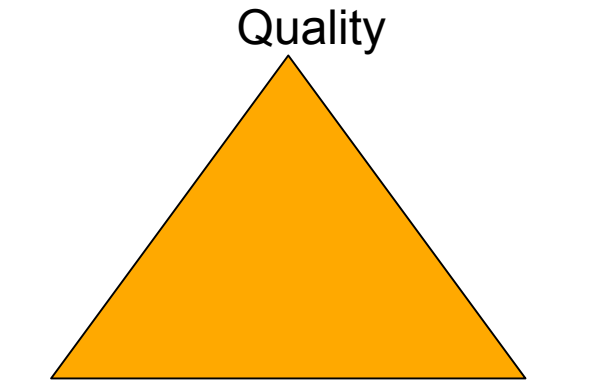
- **Vary scope** only to deliver on time and on budget
- Never compromise on quality

WATERFALL



Vary Quality

AGILE



Vary Scope

Copyright © 2003-2006 think-box Limited

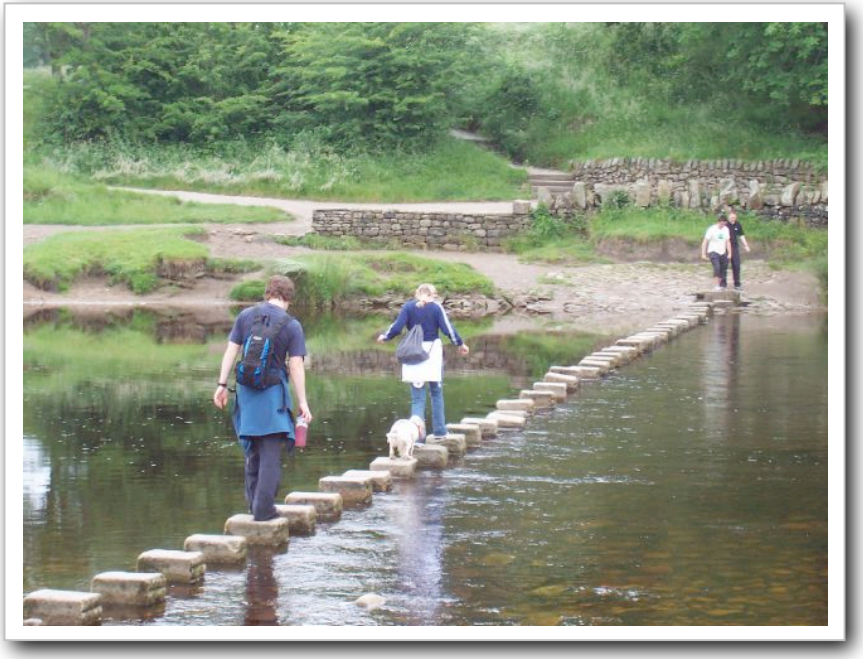
COLLABORATE DAILY

- Product Owner and Team work together every day
- **Face-to-face communication** is most effective
- Have a conversation
  - Share information
  - Seek direction
  - Achieve resolution

Copyright © 2003-2006 think-box Limited

MAKE SMALL DECISIONS

- Made quickly
- Keep you moving forward
- **Reversible**
- **Latest responsible moment** when more information is available



Copyright © 2003-2006 think-box Limited

WORKING CODE BEATS EVERYTHING

- **Running tested software (RTS)** is the best measure of progress
- Focuses attention
- Provokes more meaningful feedback
- Learn more about what you're building

Copyright © 2003-2006 think-box Limited

MAKE IT RUN. MAKE IT RIGHT. MAKE IT FAST

- Develop software iteratively
  - **Get something working**
  - Perfect it afterwards through successive refinement
- Release incrementally
  - Deliver running tested software regularly
  - **Build on what's there**
- Build quality in from the start, don't inspect for it afterwards

Copyright © 2003-2006 think-box Limited

KEEP THINGS SIMPLE

Simple rules lead to complex behaviour. Complex rules lead to stupid behaviour.

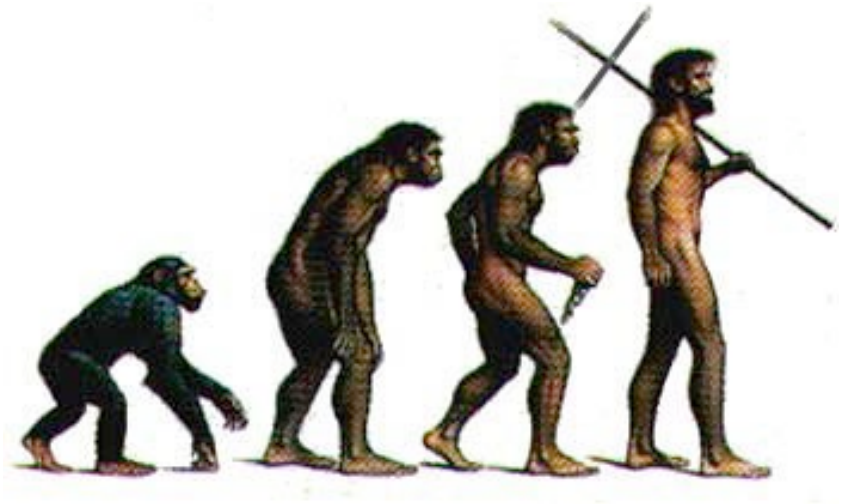
– Dee Hock, founder and former CEO of VISA

- **Simplicity** maximises the amount of work not done
- **Work by principles** and not by prescription

Copyright © 2003-2006 think-box Limited

LET THINGS EVOLVE

- Let details **emerge** as things come into focus and you get feedback
- Worry about details when they really matter
- The best architectures, requirements and designs **evolve** over time



Copyright © 2003-2006 think-box Limited

SMALL TEAMS. RIGHT PEOPLE

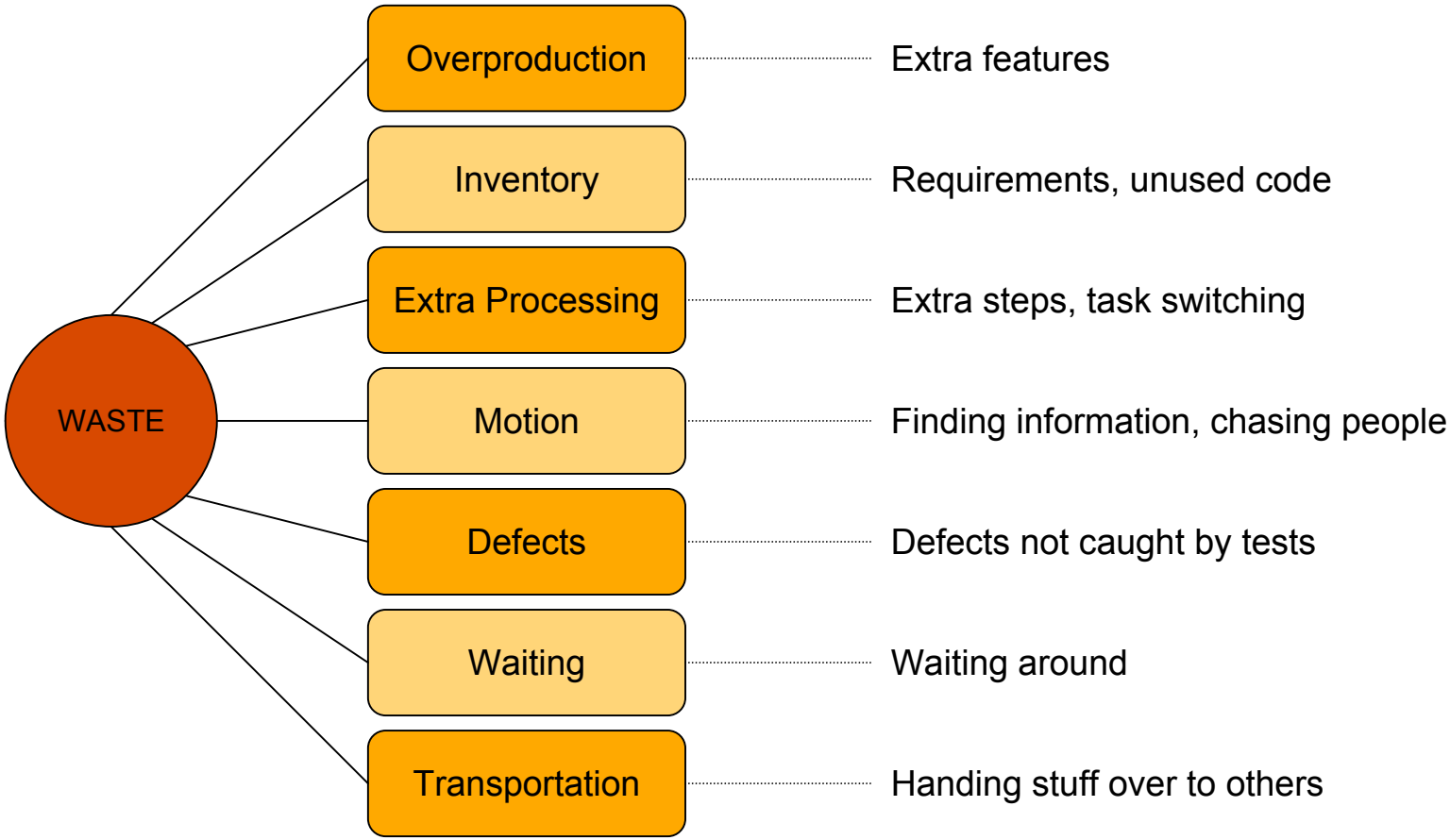
Get the right people on the bus

– Jim Collins

- Build **small teams** that can **adapt** and **respond** to change quickly
- Build projects around passionate and versatile people who are self-disciplined and who motivate themselves to do good work
- Give them the environment and support they need
- **Trust** them to get the job done

Copyright © 2003-2006 think-box Limited

ELIMINATE WASTE



Copyright © 2003-2006 think-box Limited

BE EFFECTIVE BEFORE EFFICIENT

- Don’t get so busy that you lose your ability to be effective
- Include **slack** in your capacity to give you room to manoeuvre and respond to change

Copyright © 2003-2006 think-box Limited

REFLECT TO IMPROVE

- At regular intervals, reflect on the past and how to become more effective
- Seek improvement

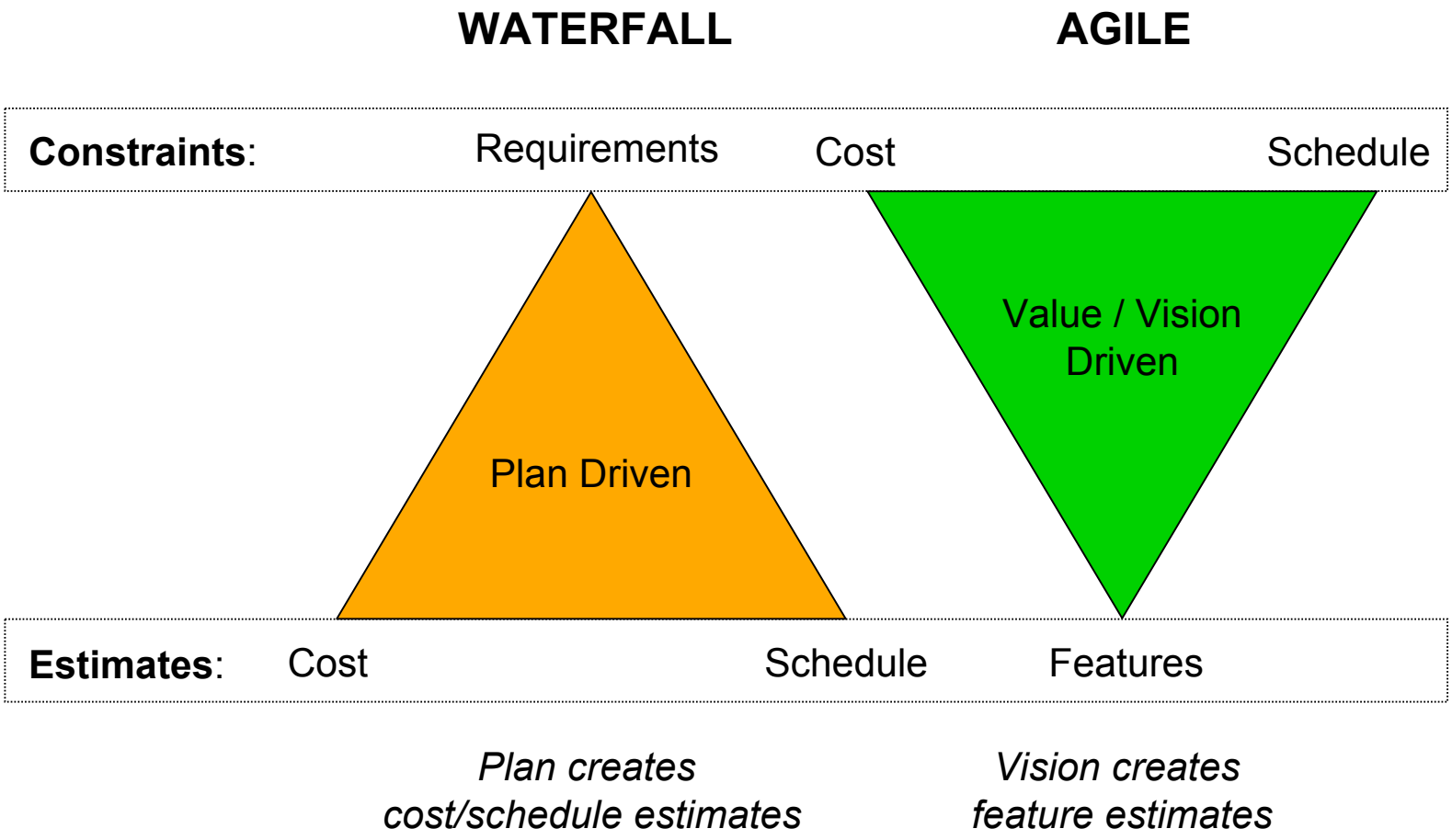
Copyright © 2003-2006 think-box Limited

AGENDA

- ☑ Waterfall [2 slides]
- ☑ Agile [21 slides]
- Basic planning concepts [5 slides]
- Scrum [23 slides]
- Scaling Scrum [8 slides]
- Next Steps [2 slides]
- Closing remarks [1 slide]

Copyright © 2003-2006 think-box Limited

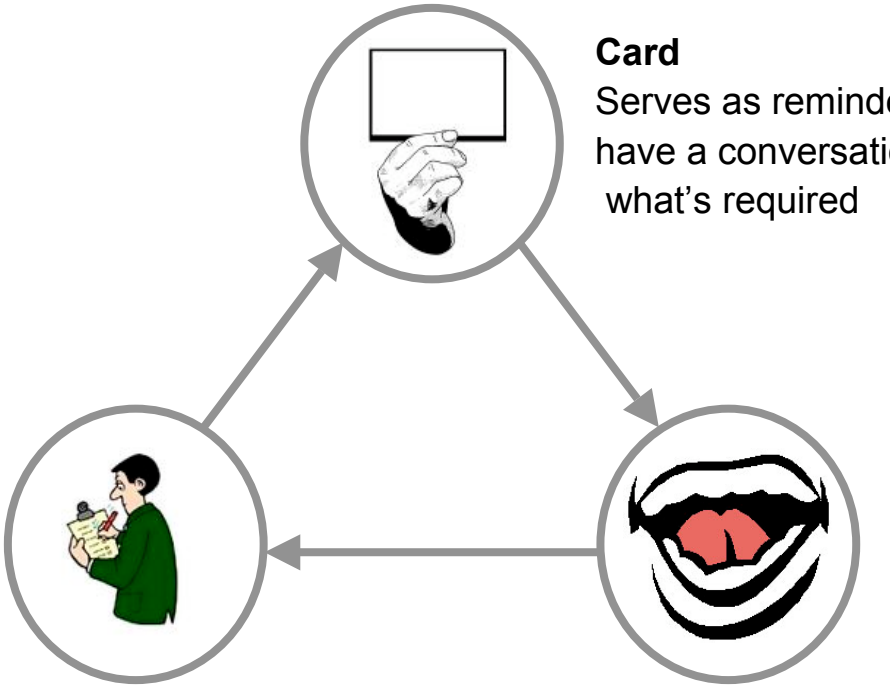
PLANNING DRIVERS



Copyright © 2003-2006 think-box Limited

USER STORIES

- Doesn't have to represent business value but must represent progress to the Product Owner
- Written in business domain language



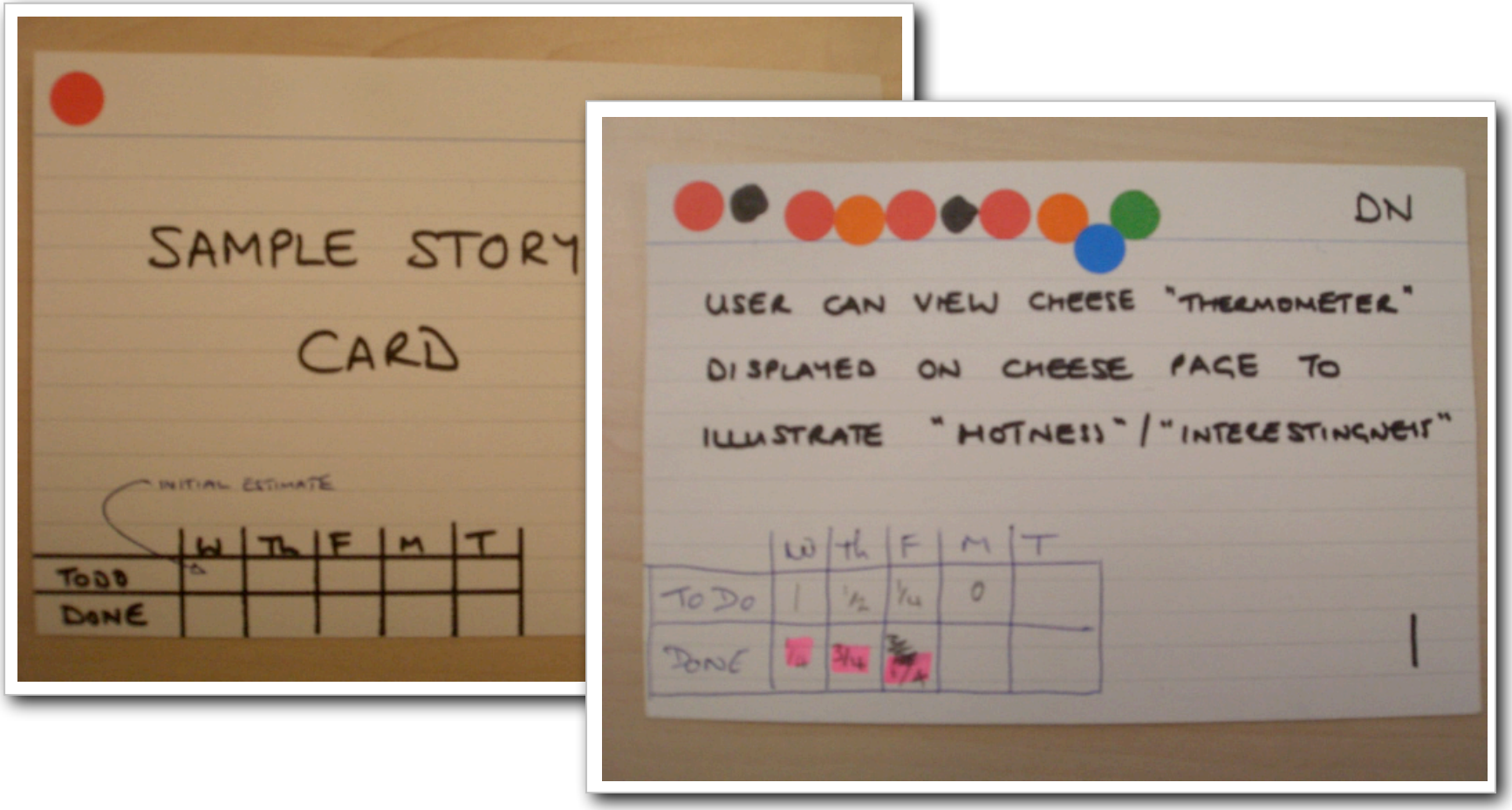
Card
Serves as reminder to have a conversation about what's required

Conversation
Collaboration and discussions about the details of the desired functionality

Confirmation
Acceptance tests capture details and are used to determine when a user story is complete

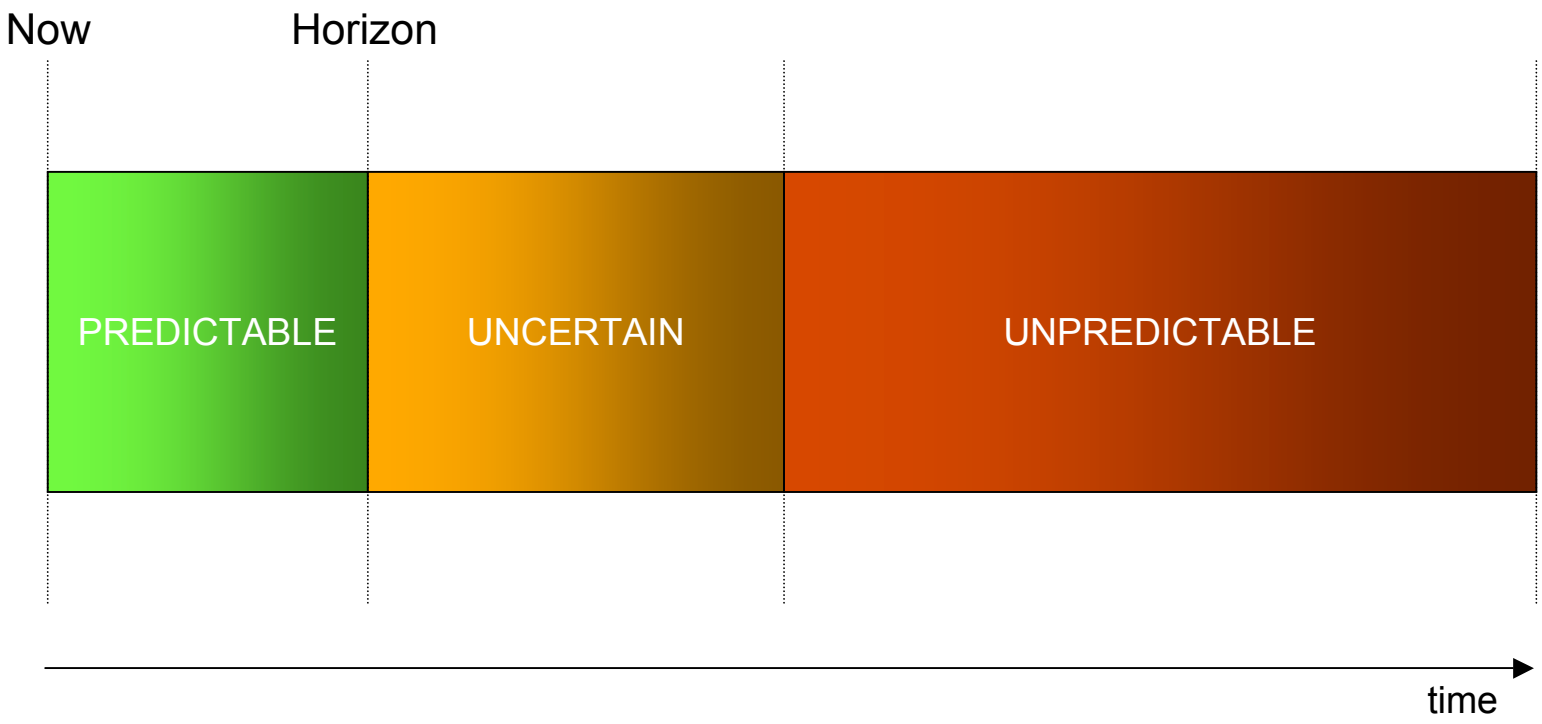
Copyright © 2003-2006 think-box Limited

STORY CARD



Copyright © 2003-2006 think-box Limited

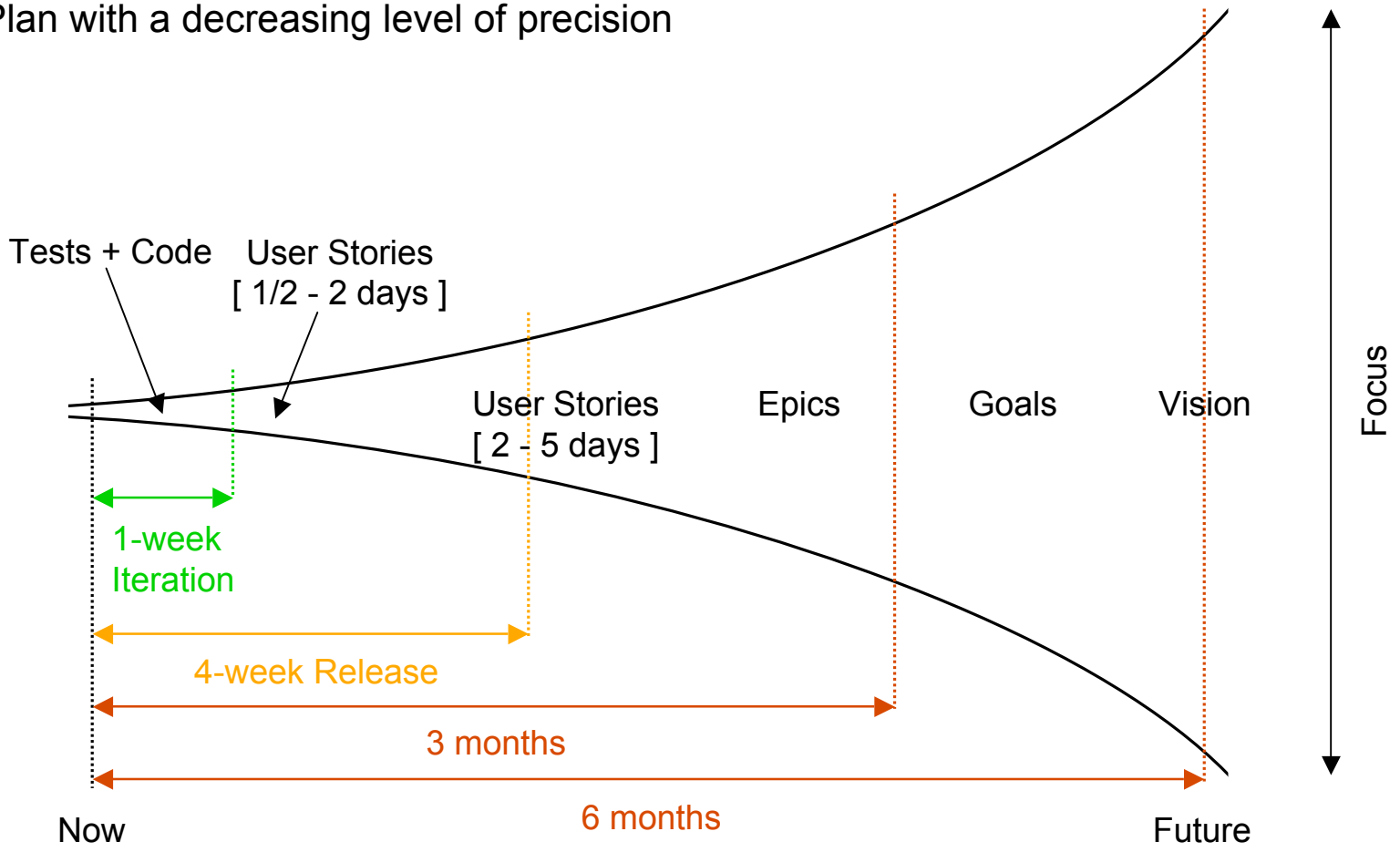
HORIZON OF PREDICTABILITY



Copyright © 2003-2006 think-box Limited

ADAPTIVE PLANNING

- Plan with a decreasing level of precision



Copyright © 2003-2006 think-box Limited

AGENDA

- ☑ Waterfall [2 slides]
- ☑ Agile [21 slides]
- ☑ Basic planning concepts [5 slides]
- Scrum [23 slides]
- Scaling Scrum [8 slides]
- Next Steps [2 slides]
- Closing remarks [1 slide]

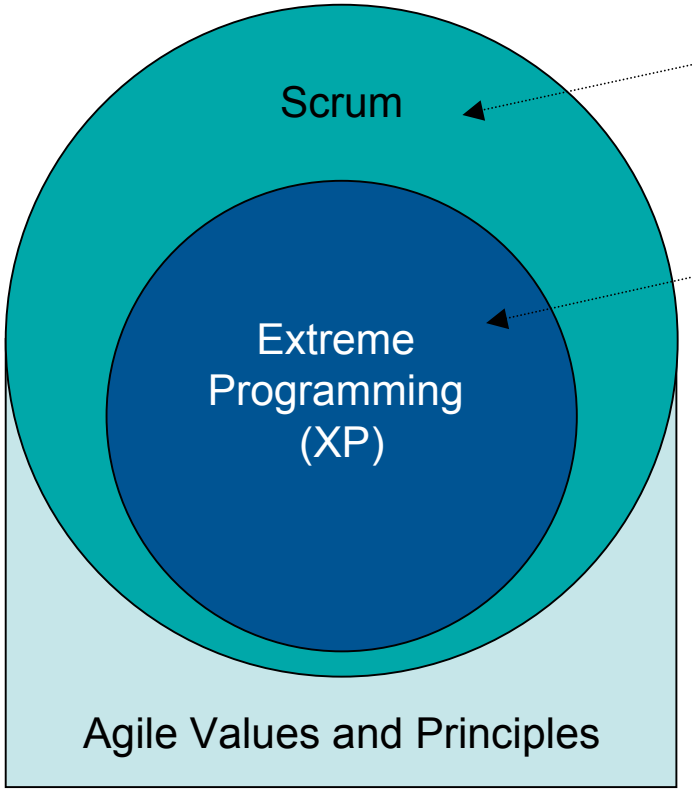
Copyright © 2003-2006 think-box Limited

SCRUM CHARACTERISTICS

- Simple and scaleable
  - Easily combined with other methods and doesn't prescribe engineering practices
- Empirical process
  - Short-term detailed planning with constant feedback provides simple inspect and adapt cycle
- Simple techniques and work artifacts
  - Requirements are captured as user stories in a list of product backlog
  - Progress is made in a series of 1 to 4-week Sprints
- Self-organising Teams collaborating with the Product Owner
- Optimises working environment
  - Reduces organisational overhead
  - Detects everything that gets in the way of delivery
- Fosters openness and demands visibility

Copyright © 2003-2006 think-box Limited

SCRUM IN CONTEXT



Facilitation techniques
Empirical process control

Software development toolkit
Engineering practices

Scrum	Extreme Programming
Product Owner	Onsite Customer
Scrum Master	N/A
Sprint	Iteration

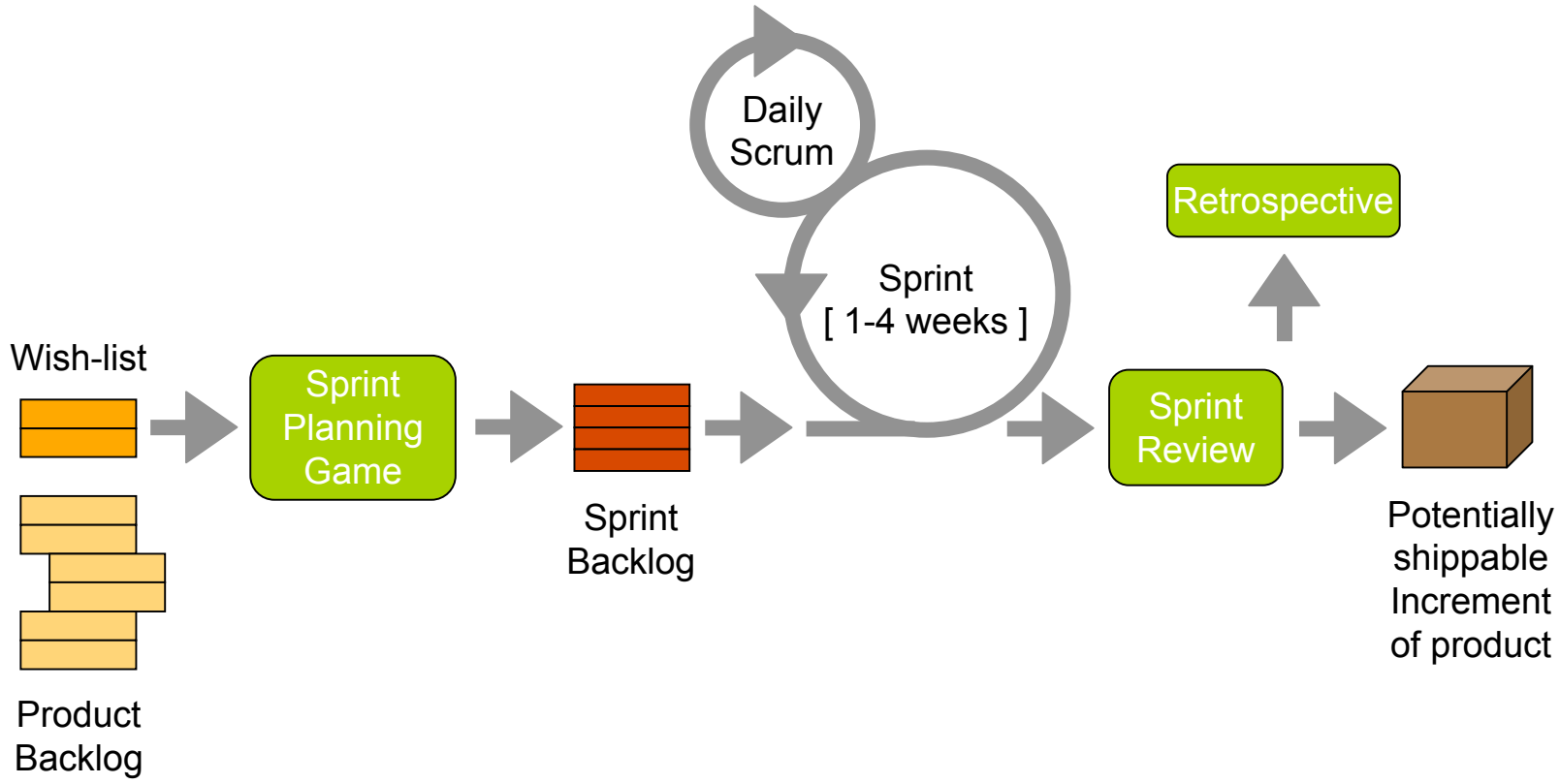
Copyright © 2003-2006 think-box Limited

EMPIRICAL PROCESS

- Visibility
  - Keep everything visible at all times
  - There's no hiding problems, issues, or obstacles
  - See them, understand them, deal with them
  - Keep working software in plain sight and in the Product Owner’s mind
- Inspection
  - Inspect frequently
  - Create as many opportunities to obtain feedback as possible
- Adaptation
  - Adapt frequently to optimise results
  - Be driven by results and not by effort
  - Measure progress by goals achieved and not by the effort it took to achieve them

Copyright © 2003-2006 think-box Limited

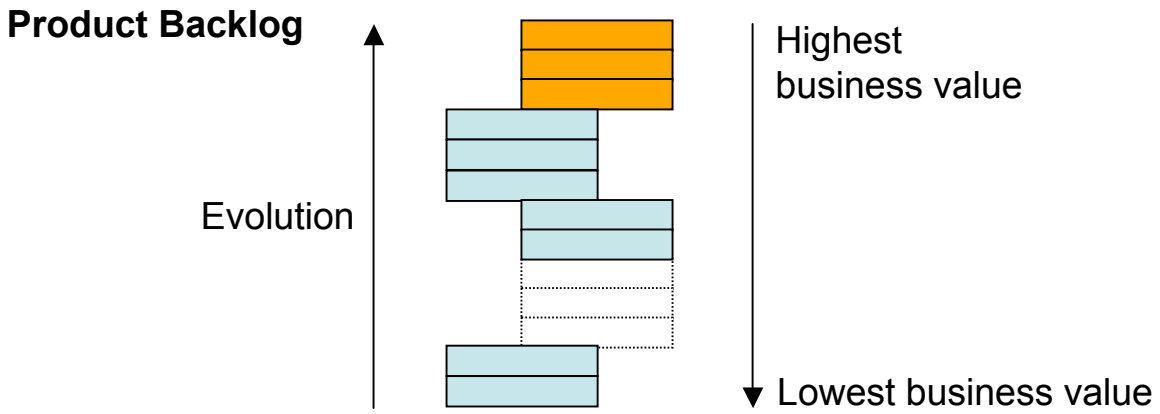
SCRUM CYCLE



Copyright © 2003-2006 think-box Limited

PRODUCT BACKLOG

- Evolving queue of work expressed as user stories
- Prioritised by business value
- Aim to deliver highest business value user stories first



Copyright © 2003-2006 think-box Limited

SPRINT

- Fixed time-box of 1-4 weeks to build something valuable for the Product Owner
- Delivers potentially shippable increment of product
- Includes development, testing, etc
- Same duration establishes rhythm

Copyright © 2003-2006 think-box Limited

SPRINT BACKLOG

- User stories planned in Sprint
- Sprint Goal
- Owned by Team
- Product Owner cannot change Sprint Goal nor Sprint Backlog once Sprint has started
- Team can:
  - Request new user stories if others completed early
  - Update estimates
  - Ask Product Owner to de-scope user stories that can't be completed
  - Terminate Sprint

Copyright © 2003-2006 think-box Limited

PRODUCT OWNER

- 1 per Team, collocated with Team
  - Ensures that only one set of requirements drives development
  - Eliminates confusion of multiple bosses, different opinions, and interference
- Visible, vocal and objective driving force making all business decisions for product
  - Responsible for business value delivered by Team
  - Reviews/accepts/rejects user stories delivered in Sprint
- Keeper/owner of the Product Backlog
  - Defines goals
  - Writes/evolves user stories narrowing their focus as they approach release/iteration planning
  - Prioritises Product Backlog by business value to set development schedule
- Attend every Release/Sprint Planning Game and Sprint Review
- Communicates user story details to Team and provides timely feedback throughout development

Copyright © 2003-2006 think-box Limited

LOOK AFTER THE PRODUCT BACKLOG

- Manage it constantly
  - Evolve it as new information becomes available
  - Stay one Sprint ahead of the Team
- Prioritise effectively to get the biggest bang for your buck
  - Identify what's valuable to the business
  - Find out the cost of each backlog item by getting provisional estimates from the Team
  - Then prioritise
- Keep the Product Backlog publicly visible, at all times

Copyright © 2003-2006 think-box Limited

SCRUM TEAM

A small number of people with complementary skills who are committed to a common purpose, performance goals, and approach for which they hold themselves mutually accountable

– Katzenbach and Smith, The Wisdom of Teams

- Small team, 7±2 full-time people, colocated
- Cross-functional with no roles, business-value oriented
  - Contains all skills required to make decisions and take action
- Self-organising, self-disciplined
  - Create and implement their own process and improvements
  - Organise themselves around the work
- Empowered, accountable
  - Authority to do whatever is needed to meet their commitments
  - Accountable for their commitments

Copyright © 2003-2006 think-box Limited

SCRUM MASTER

The Scrum Master is a sheepdog who would do anything to protect the flock, and who never gets distracted from that duty

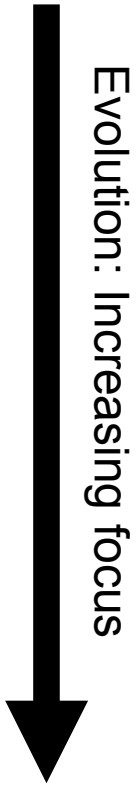
– Ken Schwaber

- 1 per Team, colocated with the Team
- Servant leader to Team
  - Facilitator without authority
  - Shows unrelenting dedication to the Team
  - Removes obstacles to help the Team maintain a sustainable pace
  - Mediates between management and the Team
- Helps Product Owner:
  - Drive the development effort directly
  - Maximise ROI
- Authority for Scrum process
  - Ensures the Team uses the values, principles and the practices

Copyright © 2003-2006 think-box Limited

5 LEVELS OF PLANNING

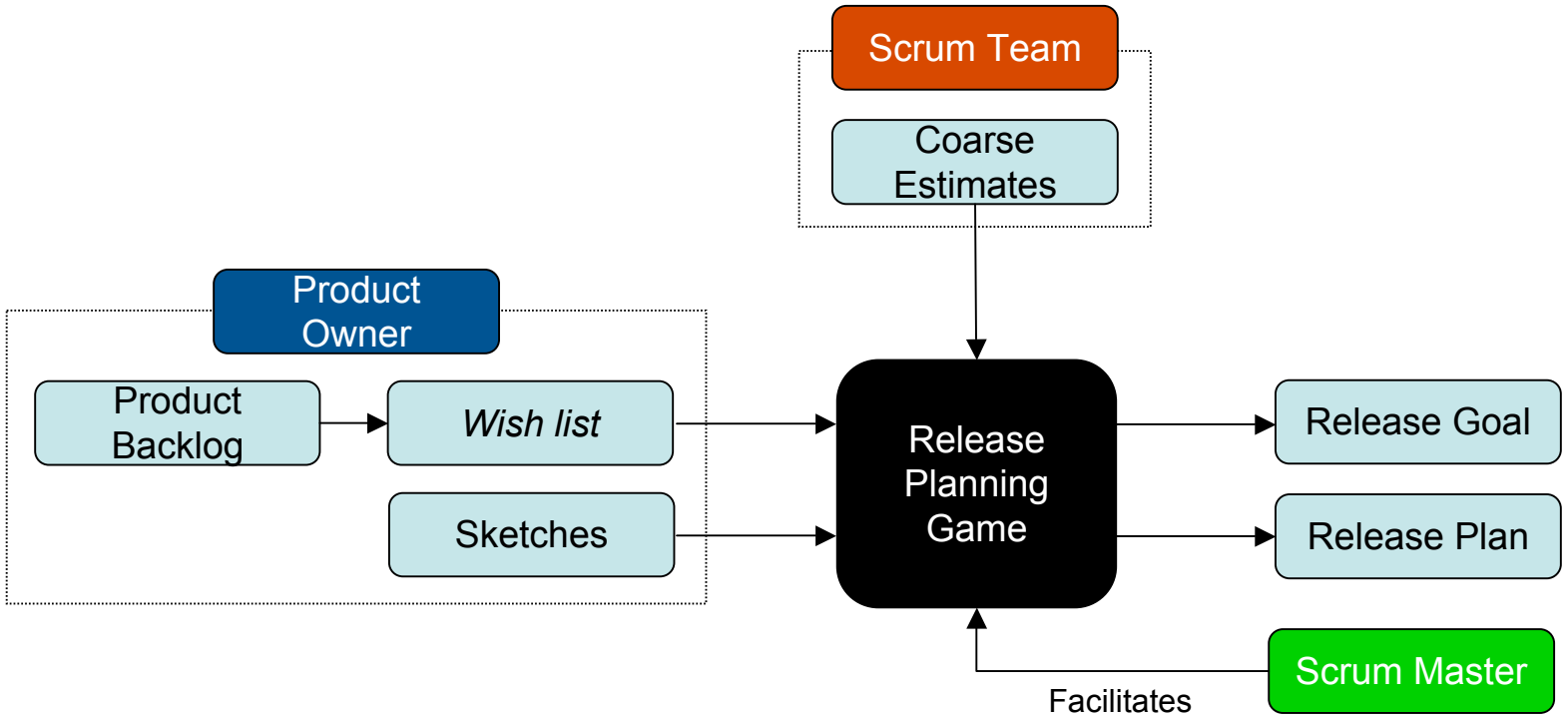
LEVEL	FREQ.	WHO	WHAT
VISION	1-2 per year	Product Owner Execs, Team	Vision Statement
ROADMAP	1-4 per year	Product Owner, Team	Product evolution over time [Goals]
RELEASE	4-12 per year	Product Owner, Team	Release Plan [User Stories]
SPRINT	Every Sprint 1-4 weeks	Product Owner, Team	Sprint Backlog [User Stories]
DAILY	Every day	Team	User Stories / Slices, Acc. Tests



Copyright © 2003-2006 think-box Limited

RELEASE PLANNING GAME

- Planning from a distance
  - Acknowledge planning can't be done with any real precision
  - Define plan that is good enough and revise it as things move forward



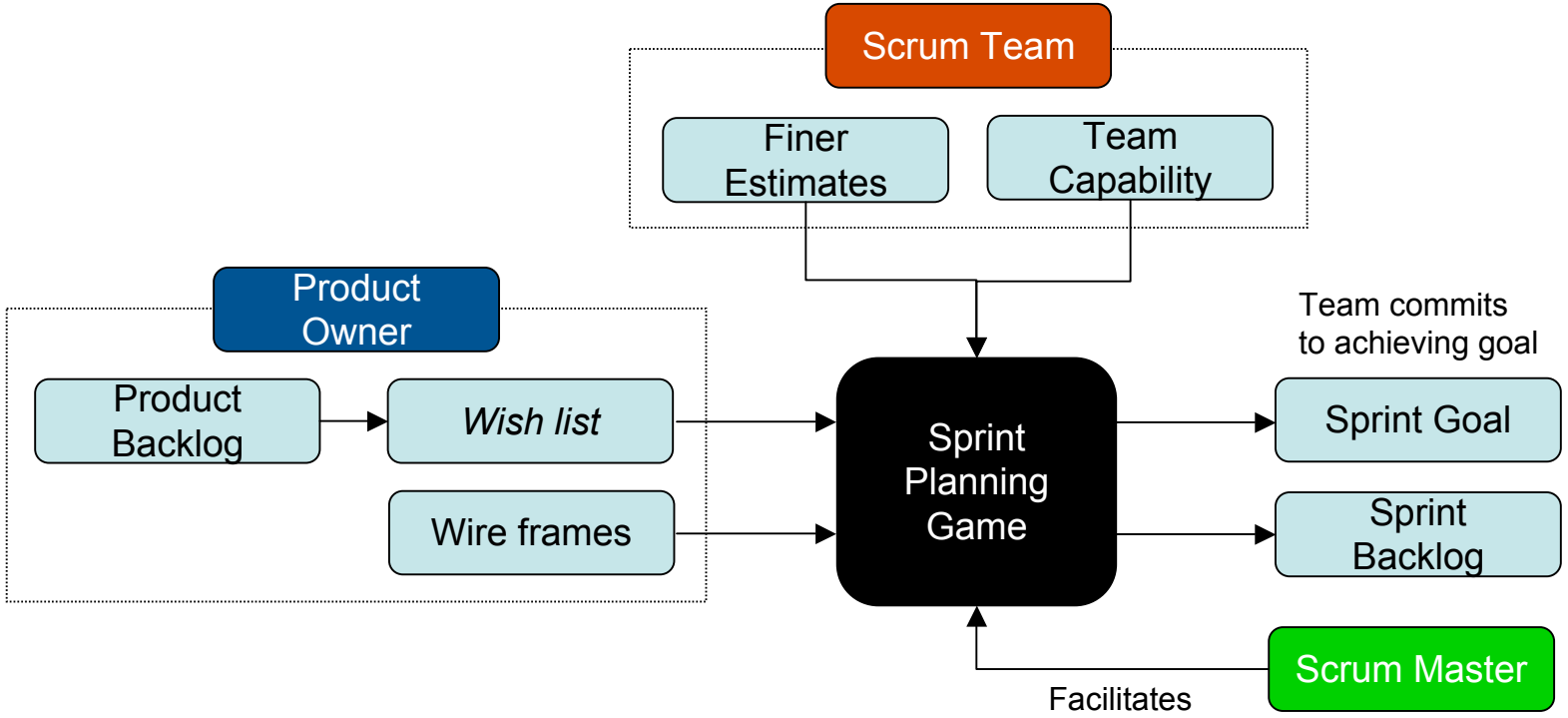
Copyright © 2003-2006 think-box Limited

PLANNING BOARD - RELEASE PLAN

Copyright © 2003-2006 think-box Limited

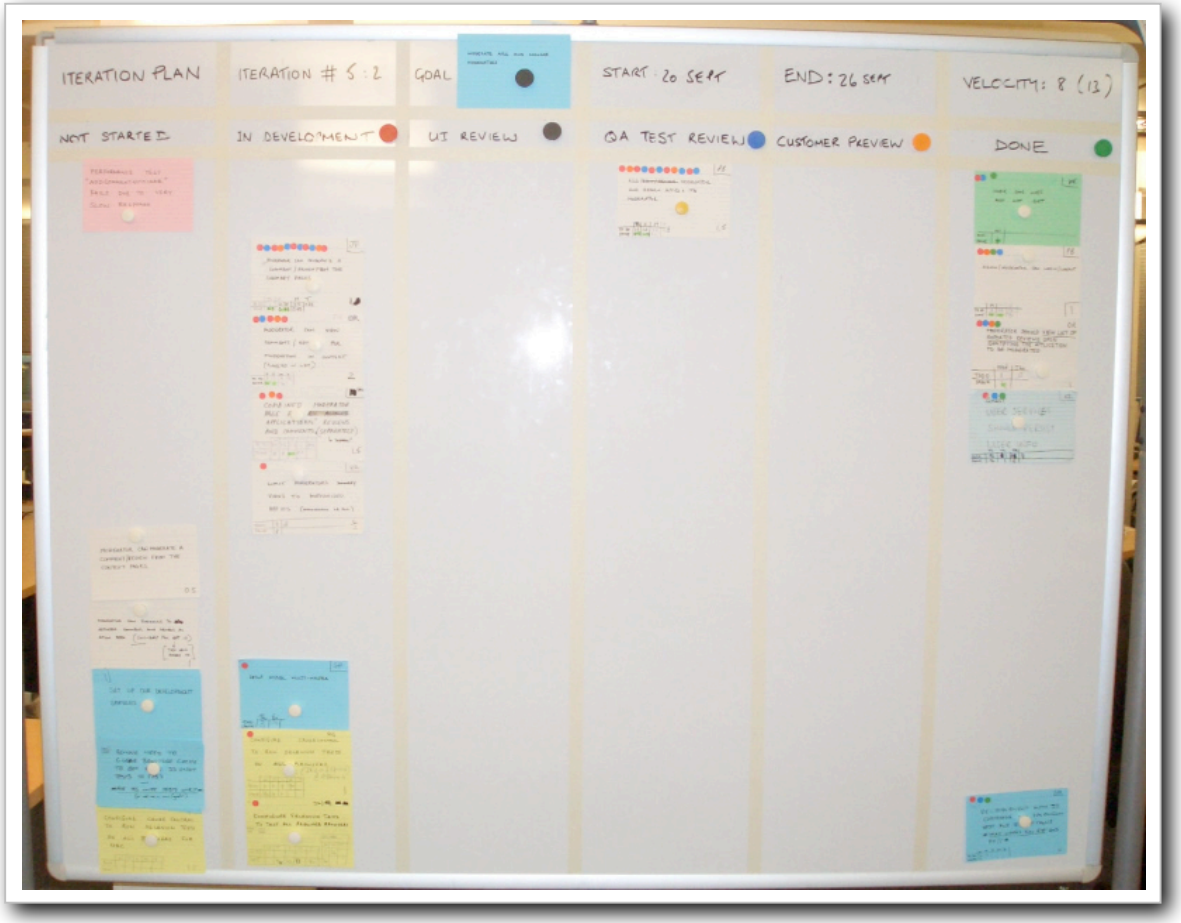
SPRINT PLANNING GAME

- Planning up close
  - Plan only next Sprint in detail when it starts
  - Use Velocity as best guess of future progress (Yesterday's Weather)



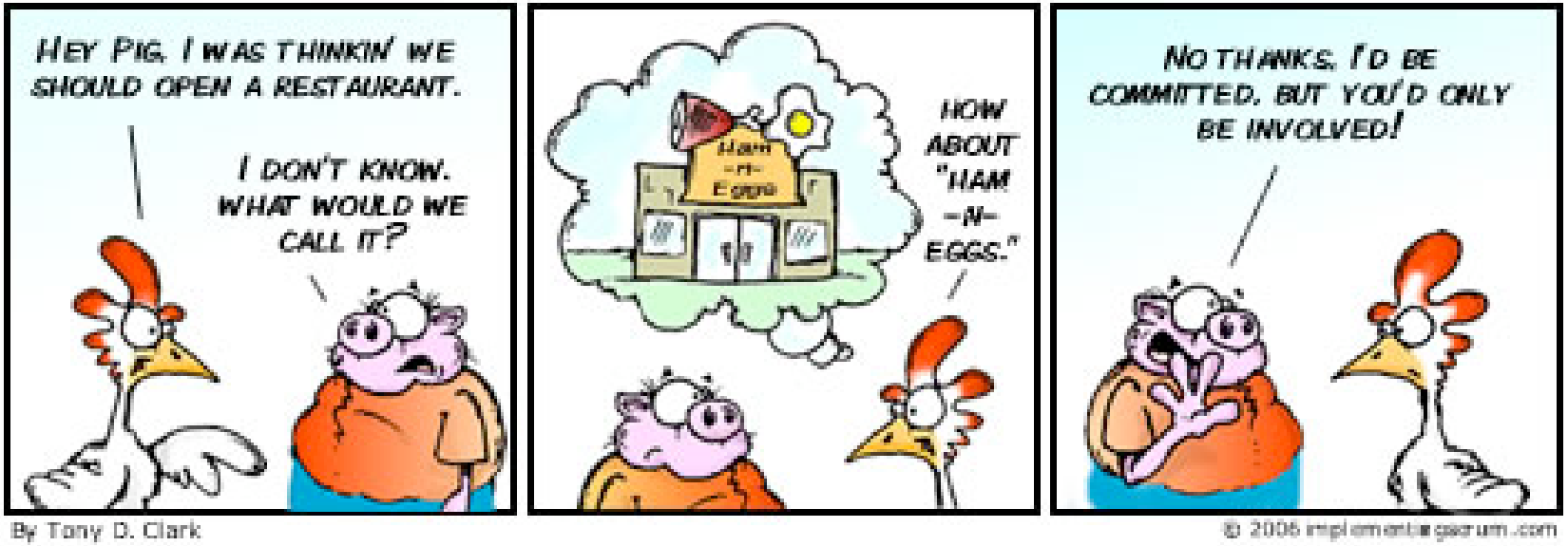
Copyright © 2003-2006 think-box Limited

PLANNING BOARD - SPRINT BACKLOG



Copyright © 2003-2006 think-box Limited

PIGS AND CHICKENS



- Scrum Team are known as pigs because they're committed to delivering Sprint Goal
- People who are involved but not dedicated to the project are known as chickens
  - Attend Daily Scrums as observers

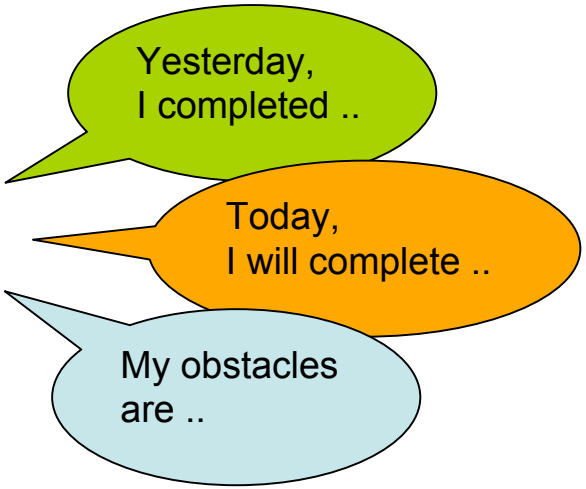
Copyright © 2003-2006 think-box Limited

DAILY SCRUM

How does a project get to be a year late? One day at a time.

– Frederick Brookes, The Mythical Man-Month

- ≤15 mins, standing up at same time every day, at same place
- Team members (pigs) talk, observers (chickens) listen
- Heartbeat of Scrum
  - Pigs co-ordinate today's work and checks progress
  - Provides daily status snapshot to chickens
- **Commitments** and **accountability**
  - Say what you'll do and do what you say
- Take discussions/problem-solving offline



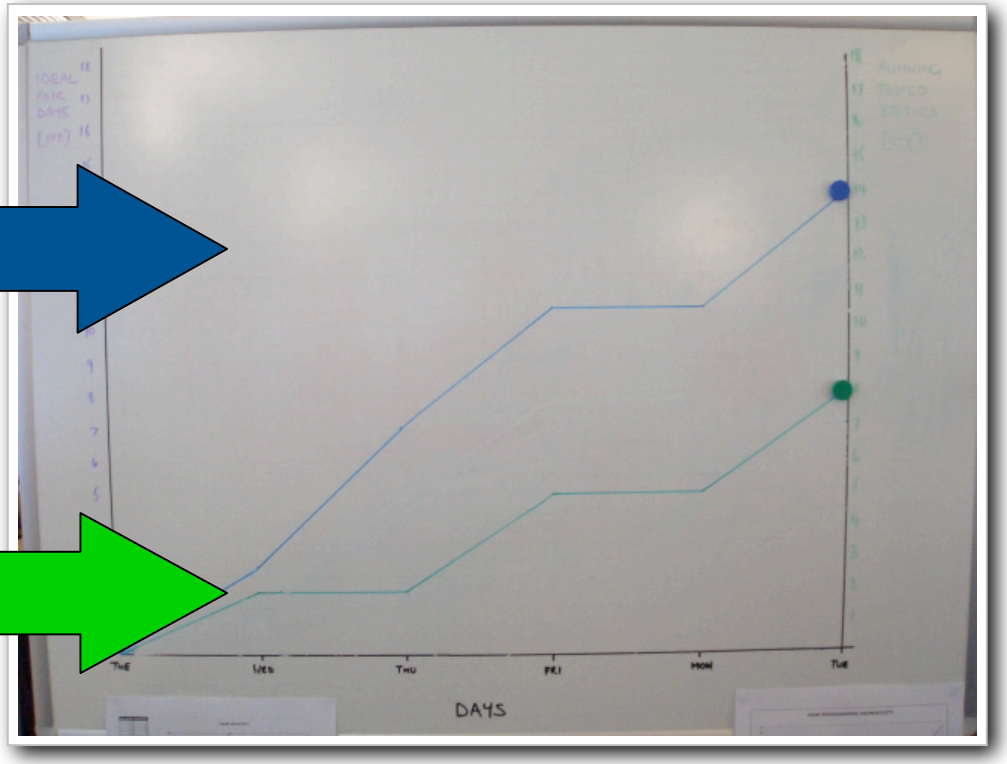
Copyright © 2003-2006 think-box Limited

TRACKING PROGRESS

- Measure real progress
  - How much more work we still have to do
  - How fast we are doing work so that we know where we're at

Burn-up chart:
Visible indication of whether Team will finish on time, early or won't finish everything they committed to

Running Tested Stories:
How many user stories are done



Copyright © 2003-2006 think-box Limited

KNOWING WHEN YOU'RE DONE

Producing production-ready quality software every sprint takes:

- Discipline
- Pursuit of excellence
- Intense and effective collaboration
- Common definition of done - A user story is 100% ready to deploy to production
  - Accepted by product owner
  - All automated unit tests pass with coverage at 85% - 100%
  - All automated acceptance tests pass
  - Any additional QA testing has passed
  - Code is checked-in, integrated and builds successfully
  - Code compiles and has a simple, well factored design without duplication
  - Code is clean and structured to coding standards
  - Code is self-documenting and clearly communicates developers' intentions

Copyright © 2003-2006 think-box Limited

SPRINT REVIEW

- ≤30 minutes at end of every Sprint
- Product Owner, Team and other Stakeholders
- Informal demonstration of functionality delivered in Sprint
- Product Owner inspects completed business value
  - Establish whether Sprint Goal has been satisfied
  - Accepts/rejects functionality delivered by user stories
  - Provide feedback
- Should feel like natural result and closure for Sprint

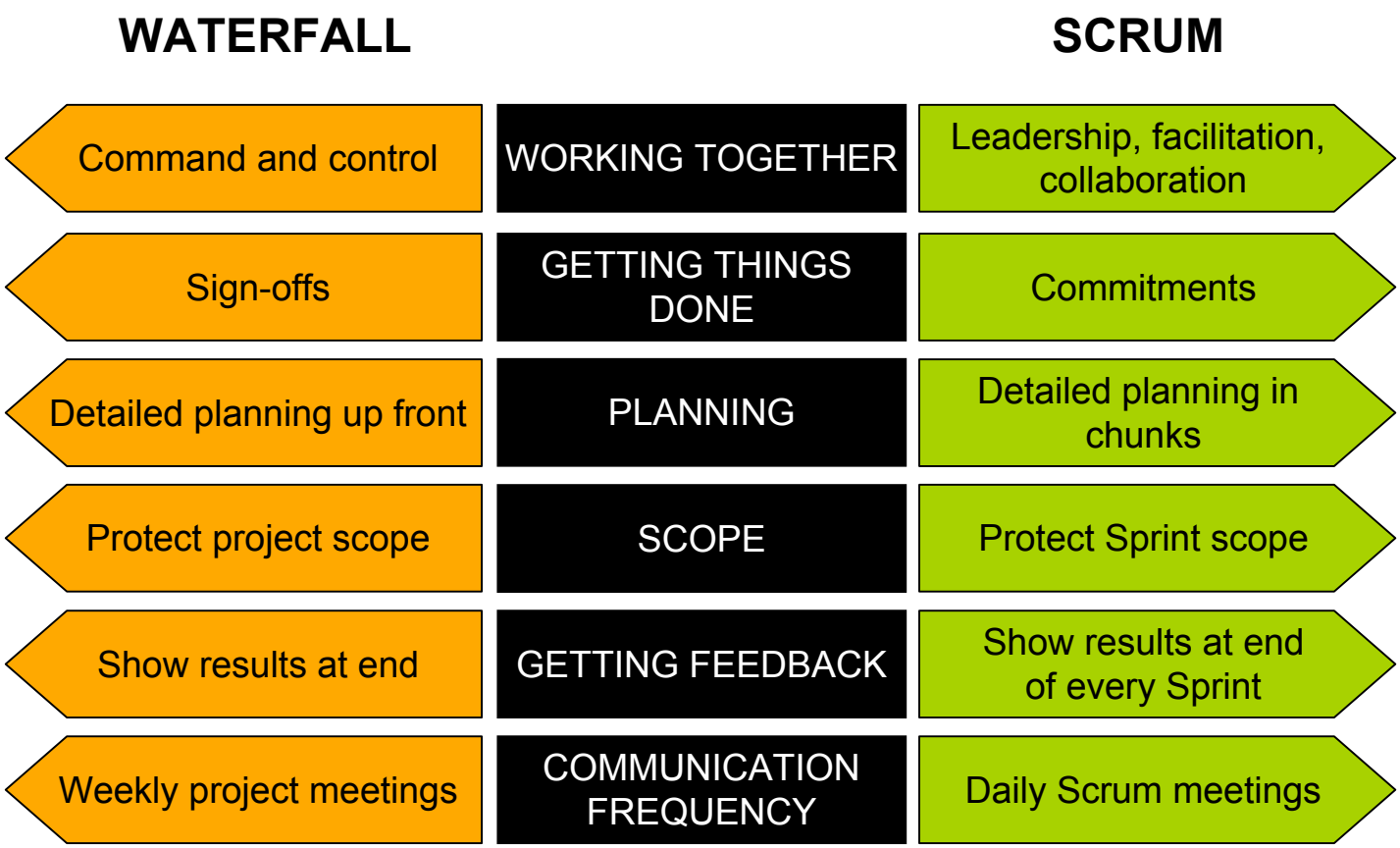
Copyright © 2003-2006 think-box Limited

RETROSPECTIVE

- Time to reflect
- Amplify learning, seek improvement and adapt
- Release retrospective
  - 2 hours - 1 day
  - Product Owner, Team, other stakeholders
  - Reflect on project, progress, alignment with roadmap
  - Identify bottlenecks and initiate repairs
- Heartbeat retrospective
  - 1 hour at end of every Sprint
  - Team
  - Reflect on process and how Team is working and initiate improvements

Copyright © 2003-2006 think-box Limited

TALE OF 2 CULTURES



Copyright © 2003-2006 think-box Limited

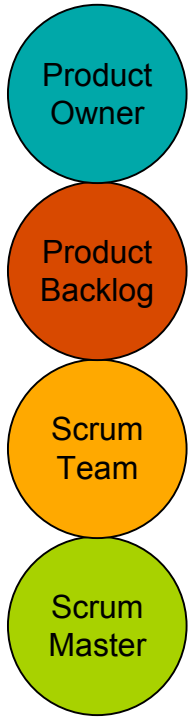
AGENDA

- ☑ Waterfall [2 slides]
- ☑ Agile [21 slides]
- ☑ Basic planning concepts [5 slides]
- ☑ Scrum [23 slides]
- Scaling Scrum [8 slides]
- Next Steps [2 slides]
- Closing remarks [1 slide]

Copyright © 2003-2006 think-box Limited

IDEAL - NO NEED TO SCALE

- Independent
- Coherent
  - Owns specific feature/service/product
- No coupling
  - Zero dependencies on other Teams to deliver business value
- Dedicated Product Owner and Product Backlog
- Dedicated Scrum Master



Product

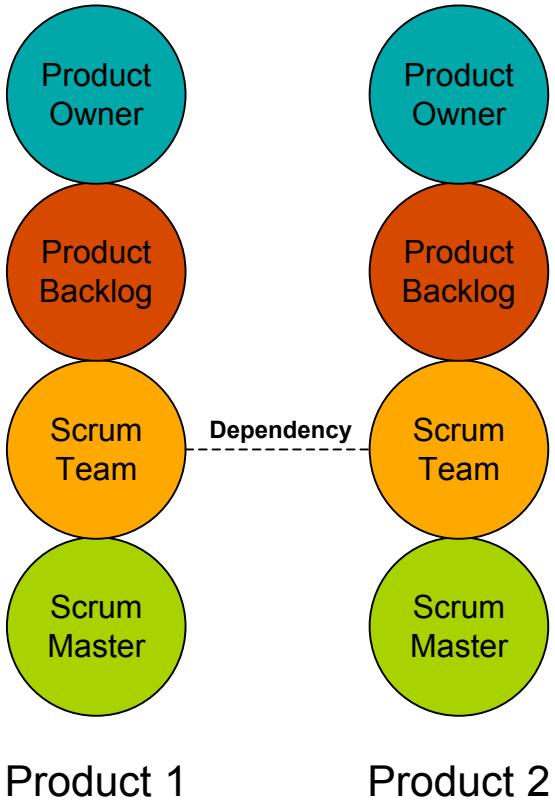
Copyright © 2003-2006 think-box Limited

WHEN TO SCALE SCRUM

- >1 Team AND dependencies between Teams

Copyright © 2003-2006 think-box Limited

COLOCATED DEPENDENCY



Advantages:

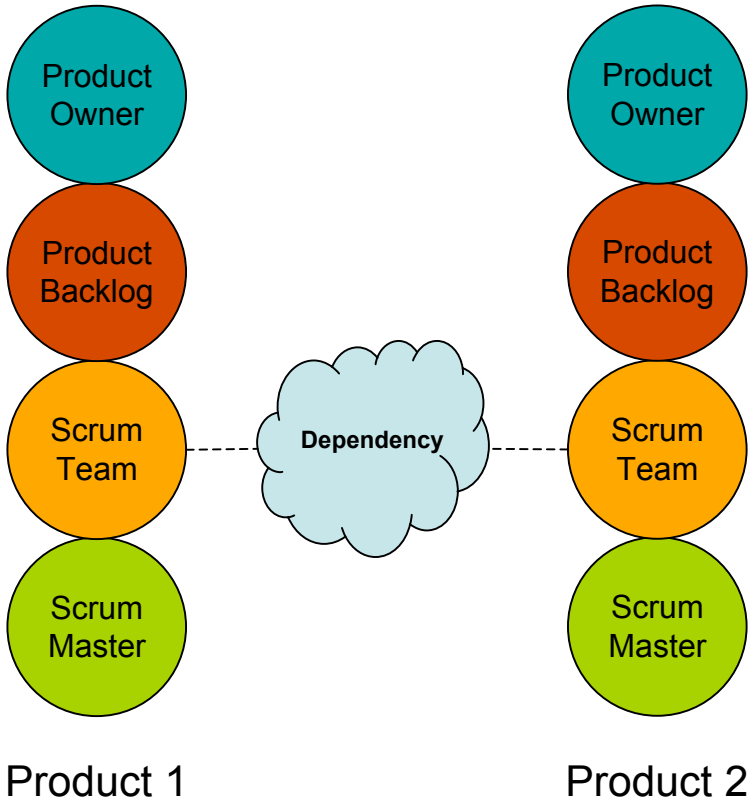
1. Visibility can be maintained
2. Face-to-face communication is possible
3. Easier to synchronise Sprints
4. Both Teams deliver incrementally
5. Integration can be better co-ordinated

Disadvantages:

1. Potential waste (motion, waiting, extra processing, transportation)

Copyright © 2003-2006 think-box Limited

DISTRIBUTED DEPENDENCY



Advantages:

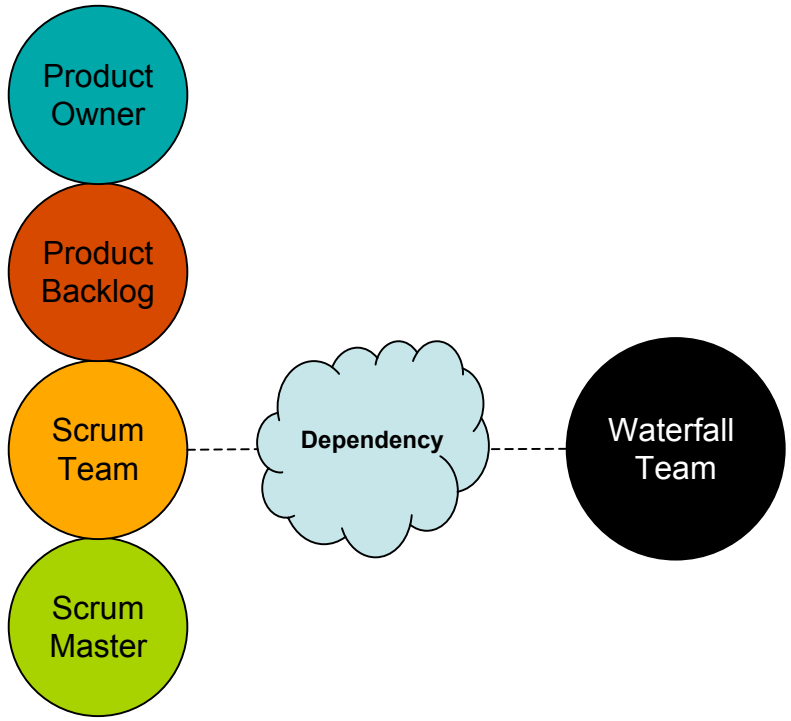
1. Both Teams value visibility and communication
2. Both Teams deliver incrementally
3. Sprints can be synchronised
4. More opportunities for integration

Disadvantages:

1. Having the dependency
2. Visibility is hindered
3. Face-to-face communication is impractical
4. Time-zones may affect real-time communication
5. Potential for waste (motion, waiting, extra processing, transportation)

Copyright © 2003-2006 think-box Limited

DISTRIBUTED DEPENDENCY



Advantages:

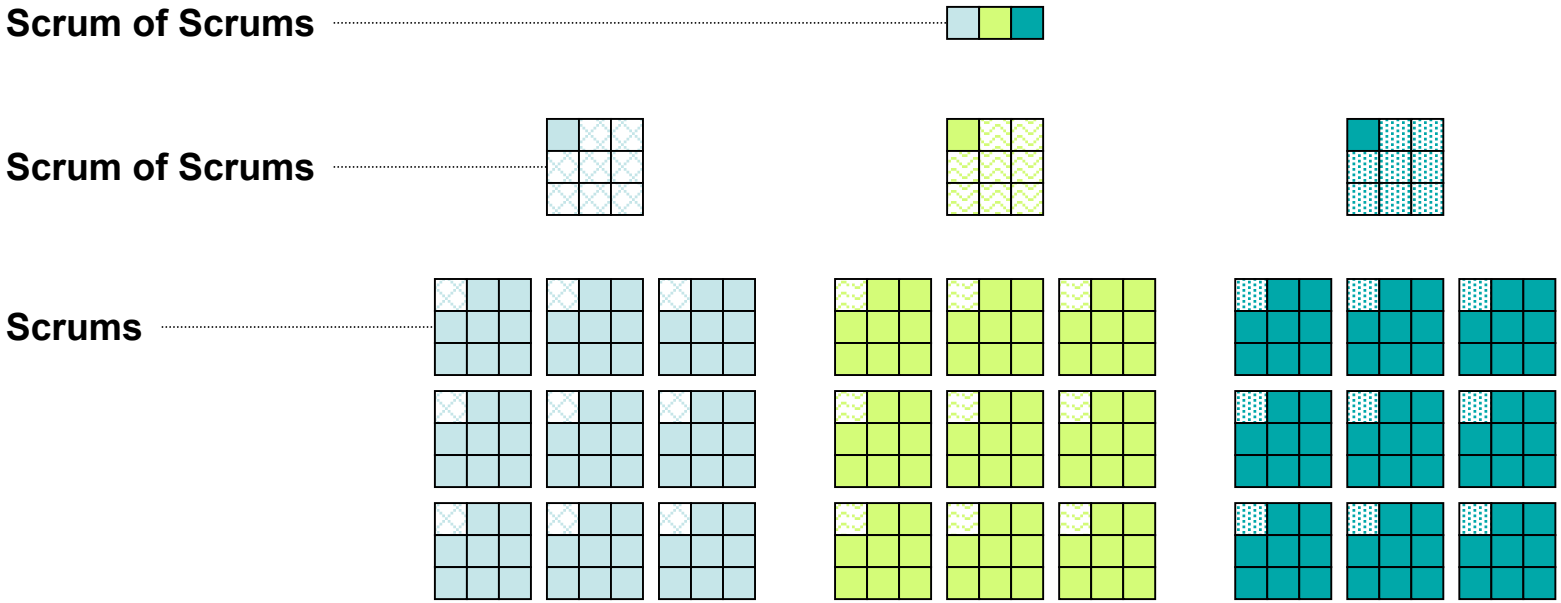
Disadvantages:

1. Visibility is seriously hindered
2. Face-to-face communication is impractical
3. Time-zones may affect real-time communication
4. Fewer opportunities for integration
5. Enormous potential for waste (motion, waiting, extra processing, transportation)
6. Waterfall team will not go to extremes to communicate and make everything visible
7. Waterfall team is not adaptable

Copyright © 2003-2006 think-box Limited

SCRUM OF SCRUMS

- ≤ 1 hour, at the same time, daily or every other day
- Attended by 1 person (pig) from each Team
  - Team should regularly change the person who attends



Copyright © 2003-2006 think-box Limited

SCRUM OF SCRUMS

- Communicate progress and obstacles among Teams
- Resolve co-ordination and dependency problems
- Synchronise Sprints
- Measure impact and adjust Sprints accordingly
- Work from a different backlog
  - Issues to resolve
  - Actions to take
  - Decisions to make
  - Obstacles to remove

Copyright © 2003-2006 think-box Limited

WORKING AROUND DISADVANTAGES

1. Visibility is hindered — Use online applications: Wiki, agile planning tool
2. Face-to-face communication is impractical — Video-conferencing or Web cam with IM, Skype
3. Time-zones may affect real-time communication — Batch up communications in email
4. Fewer opportunities for integration — Buffering, interface-driven, mocking
5. Potential for waste — ?
6. Waterfall team is not adaptable and will not go to extremes to communicate and make everything visible — Ask them nicely to change their working practices

Copyright © 2003-2006 think-box Limited

AGENDA

- ☑ Waterfall [2 slides]
- ☑ Agile [21 slides]
- ☑ Basic planning concepts [5 slides]
- ☑ Scrum [23 slides]
- ☑ Scaling Scrum [8 slides]
- Next Steps [2 slides]
- Closing remarks [1 slide]

Copyright © 2003-2006 think-box Limited

REORGANISE FOR SCRUM

1. Corporate/Exec/Program Office, top-down:
   a) Communicate intention and reasons for adopting agile methods more broadly
   b) Communicate adoption plan
   c) Begin to demonstrate buy-in to values and principles
   d) Focus attention on incremental delivery of running tested software
   e) Request progress reporting in terms of running tested software

2. Business/Tech, bottom-up:
   a) Identify business value streams and find Product Owners
   b) Build Teams, not groups of people
   c) Build Teams organised around features/services/products
   d) Hire Scrum Masters not Project Managers
   e) Provide coaching

Copyright © 2003-2006 think-box Limited

GETTING STARTED WITH SCRUM

1. Product Owner captures Product Vision
2. Product Owner sets out a Goal Roadmap
3. Product Owner creates initial Product Backlog
4. Set-up new Teams
   - Bull-pen, informative workspace
   - Planning board
   - Run an *Iteration Zero*
5. Product Owners/Teams plan Release/Sprint
6. Teams start synchronised Sprints

Copyright © 2003-2006 think-box Limited

AGENDA

- ☑ Waterfall [2 slides]
- ☑ Agile [21 slides]
- ☑ Basic planning concepts [5 slides]
- ☑ Scrum [23 slides]
- ☑ Scaling Scrum [8 slides]
- ☑ Next steps [2 slides]
- Closing remarks [1 slide]

Copyright © 2003-2006 think-box Limited

CLOSING REMARKS

Choosing to adopt agile methods is the start of an extensive change process.

– Bob Schatz, Primavera

Adopting agile methods is a commitment to an effort that involves everyone

– Bob Schatz, Primavera

Copyright © 2003-2006 think-box Limited

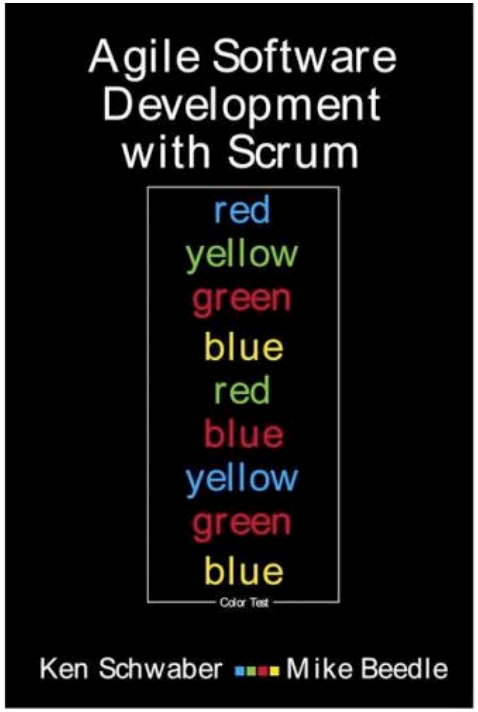
AGENDA

- ☑ Waterfall [2 slides]
- ☑ Agile [21 slides]
- ☑ Basic planning concepts [5 slides]
- ☑ Scrum [23 slides]
- ☑ Scaling Scrum [8 slides]
- ☑ Next steps [2 slides]
- ☑ Closing remarks [1 slide]

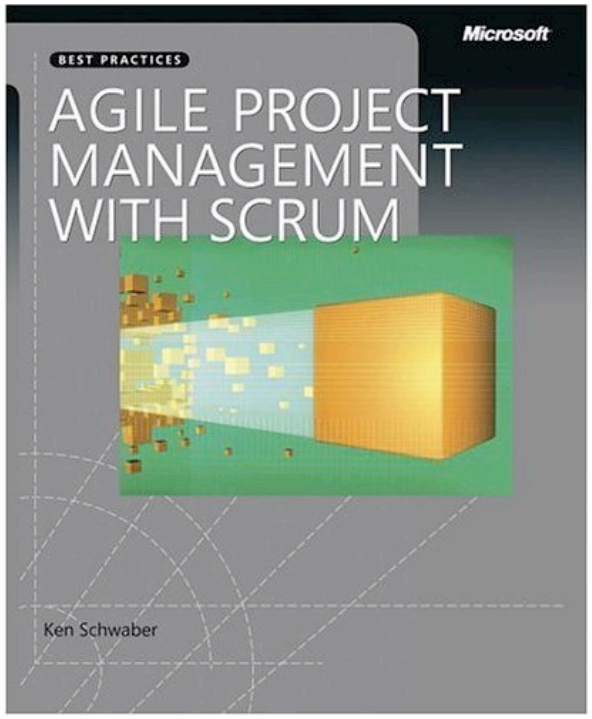
Copyright © 2003-2006 think-box Limited

REFERENCES AND FURTHER READING

Agile Software Development with Scrum
Schwaber K, Beedle M (2002)
Prentice Hall



Agile Project Management with Scrum
Schwaber K (2004)
Microsoft Press



Copyright © 2003-2006 think-box Limited

CONTACT DETAILS

- Email: simonbaker@think-box.co.uk
- Blog: http://www.agileinaction.com
- Website: http://www.think-box.co.uk

Copyright © 2003-2006 think-box Limited