

IFB299 Application Design and Development Week One

Lecturer: Prakash Bhandari

Agenda

- ◆ Introduction to IFB299
- ◆ Agile & Scrum
- ◆ User Stories

Intro to IFB299

- Application Design and Development
- Teaching Team
- Assumptions
- Workshops
- Assessment

Application Design & Development



Teaching Team

- ◆ Unit Coordinator

- ◆ Prakash Bhandari

- ◆ email:

- p.bhandari@qut.edu.au

- ph: 3138 1927

- ◆ office: S1210

- ◆ behind a locked door

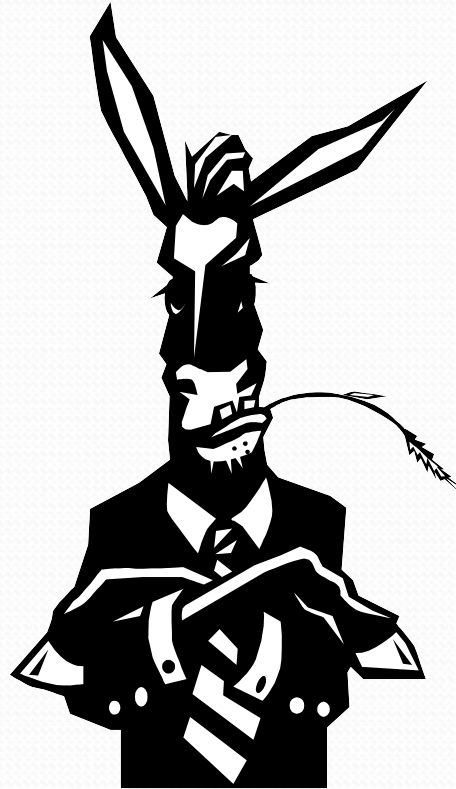
- ◆ call me to let you in

Tutors

- ◆ Mr. Andrew Hyland
Email: andrew.hyland@qut.edu.au
- ◆ Mr. Sina Aminmansour
Email: s.aminmansour@qut.edu.au
- ◆ Ms. Afsaneh Bahmani
Email: afsaneh.ghasemighalehbahmani@qut.edu.au
- ◆ Mr. Artem Kulakov
Email: artem.kulakov@qut.edu.au
- ◆ Dr. Venkat Venkatachalam
Email: venkat.venkatachalam@qut.edu.au
- ◆ Mr. Syed Zaidi
Email: syedrehan.abbaszaidi@qut.edu.au

Unit Assumptions

- ◆ Students
- ◆ Agile



Student Assumptions

- ◆ You can analyse a problem & identify solutions
 - ◆ you understand the design process of selecting a solution
- ◆ You can learn new tools and techniques on your own
 - ◆ JIRA, Automatic Testing tools (Selenium), Design Tools
- ◆ You have at least basic programming ability

Agile Assumptions

- ◆ Everyone contributes
 - ◆ to everything
- ◆ Specialist roles were always a bad idea
 - ◆ Analyst, Designer, Programmer, Tester, QA, ...
 - ◆ it's an iterative process, you can't assign roles based on phases
- ◆ Specialist skills are valuable
 - ◆ no one can know everything
 - ◆ DB design, telecoms programming, testing, tools, ...

Unit Schedule & Its Demands



Workshops

- ◆ 110 minutes in length
 - ◆ Tightly timeboxed
- ◆ *Entire* team **must** show up for *all* of the workshop
- ◆ Activities centre around project development
- ◆ Each team will play developer & client roles

Good Workshops

- ◆ Come prepared
- ◆ Make use of time
 - ◆ interacting with client
 - ◆ working on project
 - ◆ getting advice from tutor
- ◆ Show artefacts
 - ◆ stories, estimates, code, tests, ...

Projects

- ◆ Community Organization
 - ◆ Meditation Centre
 - ◆ Music School
 - ◆ Package Delivery
 - ◆ Parking Permits and Violations
-
- ◆ Developers pick preferred project
 - ◆ Client randomly assigned

Platform

- ◆ Your choice
 - ◆ Standalone, Web, Mobile, Cloud
 - ◆ combination of these
- ◆ Make use of experience within your team
- ◆ Choose based on what is easier to implement vs. long term extensibility or robustness

Assessment

- ◆ It's all about the Project ...
- ◆ Project Deliverables and Process (60%)
- ◆ Development Activities (20%)
- ◆ Peer Review (20%)

Project Deliverables & Process

- ◆ Project Quality (30%)
 - ◆ Three deliverables
 - ◆ User Stories and Release & Iteration Plans (10%)
 - ◆ before End of Week 5 Workshop Day
 - ◆ Release One (10%)
 - ◆ before End of Week 9 Workshop Day
 - ◆ Release Two (10%)
 - ◆ before End of Week 13 Workshop Day
- ◆ Personal Portfolio (30%)
 - ◆ Releases 1 & 2, before End of Week 13 Workshop Day

Development Activities

- ◆ Reflective Journal (10%)
 - ◆ **Three Blog entries made as follow:**
 - ◆ First five weeks
 - ◆ Release 1 Activities (Sprint 1, Sprint 2, Demo, Peer Review etc.)
 - ◆ Release 2 Activities (Sprint 3, Sprint 4, Demo, Peer Review etc.)
 - ◆ **reflect on team effectiveness**
 - ◆ **reflect on personal contributions**
- ◆ Peer & Tutor Observations (10%)
 - ◆ Week 13 (End of semester)

Peer Review

- ◆ As a client you will provide feedback to your developers
 - ◆ Are they achieving project goals?
 - ◆ Are they achieving technical excellence?
- ◆ Assessed on
 - ◆ Quality of feedback
 - ◆ Effectiveness as a client
- ◆ Weeks 7 & 11

Weekly Schedule

	Lecture	Workshop	Assessment Due
Week 1	Introduction to Scrum & Project	Form Teams & Choose Project	
Week 2	Stakeholders, Scope & Specifications	Start Requirements Specification	
Week 3	Priority, Estimation & Planning	Refine Requirements & Start Estimation	
Week 4	System Implementation and Architecture	Refine Requirements & Start Prioritisation	
Week 5	Database Design - I	Finalise Priorities & Sprint Planning	User Stories & Plans
Week 6	Database Design - II	Mid-Iteration Demo	
Week 7	System Testing	Sprint Review & Sprint Planning	Peer Review
Week 8	User Interface and Design	Mid-Iteration Demo	
Week 9	Conflict Resolution	Sprint Review & Sprint Planning	First Release
Week 10		Mid-Iteration Demo	
Week 11		Sprint Review & Sprint Planning	Peer Review
Week 12		Mid-Iteration Demo	Reflective Journal
Week 13		Sprint Review & Final Demo	Final Release , Portfolio, Peer & Tutor Review

Communication Mechanisms

- ◆ Communication among Teaching Staff and Student(s)
 - ◆ Frequently Asked Questions (FAQ) document
 - ◆ Available from Blackboard
- ◆ Tutor(s)
- ◆ Unit Coordinator
 - ◆ Send an email or ask for meeting via email

Agile & Scrum

- Agile
- Scrum

Agile



lynchr.co.uk



Why Agile?

- ◆ Who wants to be rigid? 😊
- ◆ User / Customer focussed
 - ◆ early, continuous & frequent delivery
 - ◆ quick & continuous feedback
- ◆ Becoming a mainstream PM method

Values of Agile Development

**Individuals and
Interactions**

over

Process and Tools

Working Product

over

**Comprehensive
Documentation**

**Customer
Collaboration**

over

Contract Negotiation

**Responding to
Change**

over

Following a Plan

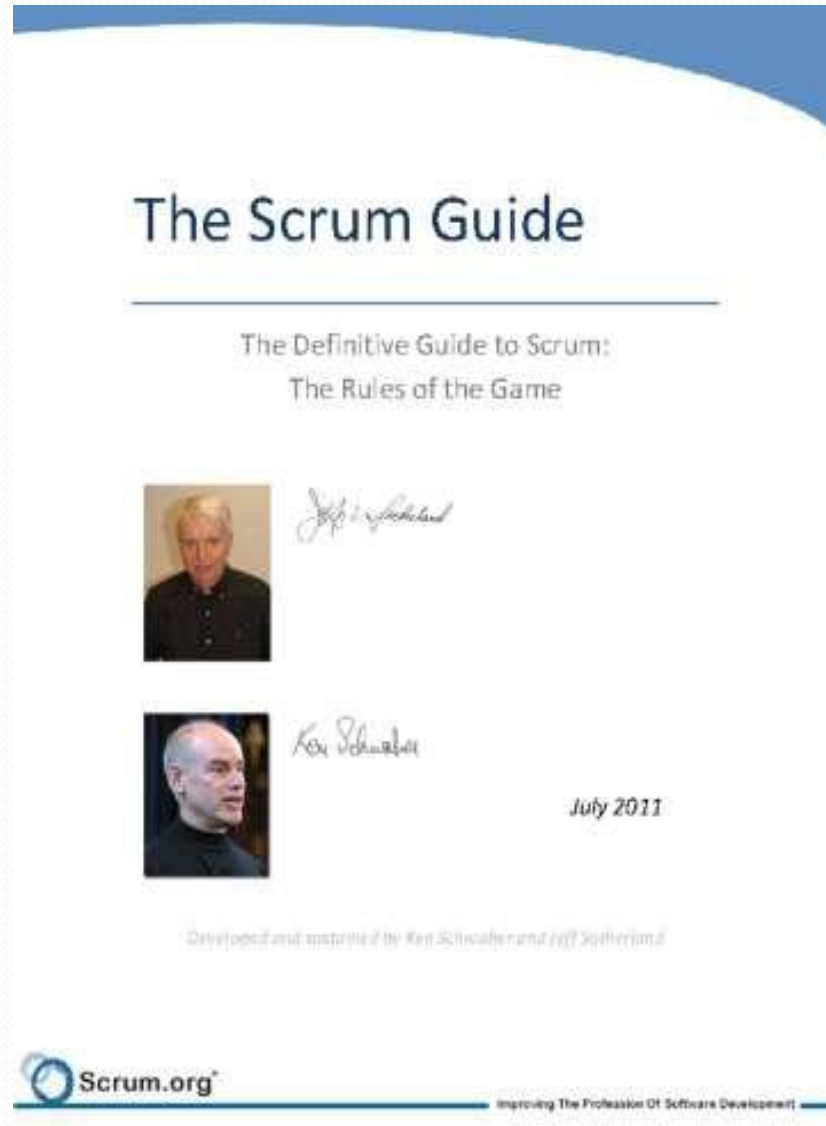
Some Agile Wisdom

- ◆ Embrace change!
 - ◆ Requirements are never, ever, ever, fixed
 - ◆ Stop pretending, and get used to it
- ◆ Deliver early and deliver often
 - ◆ A working system delivers value
 - ◆ telephone book scale documentation does not
 - ◆ A deployed system generates revenue
 - ◆ 80:20 rule

Scrum



Reading



<http://www.scrumguides.org/>

Companies Using Scrum

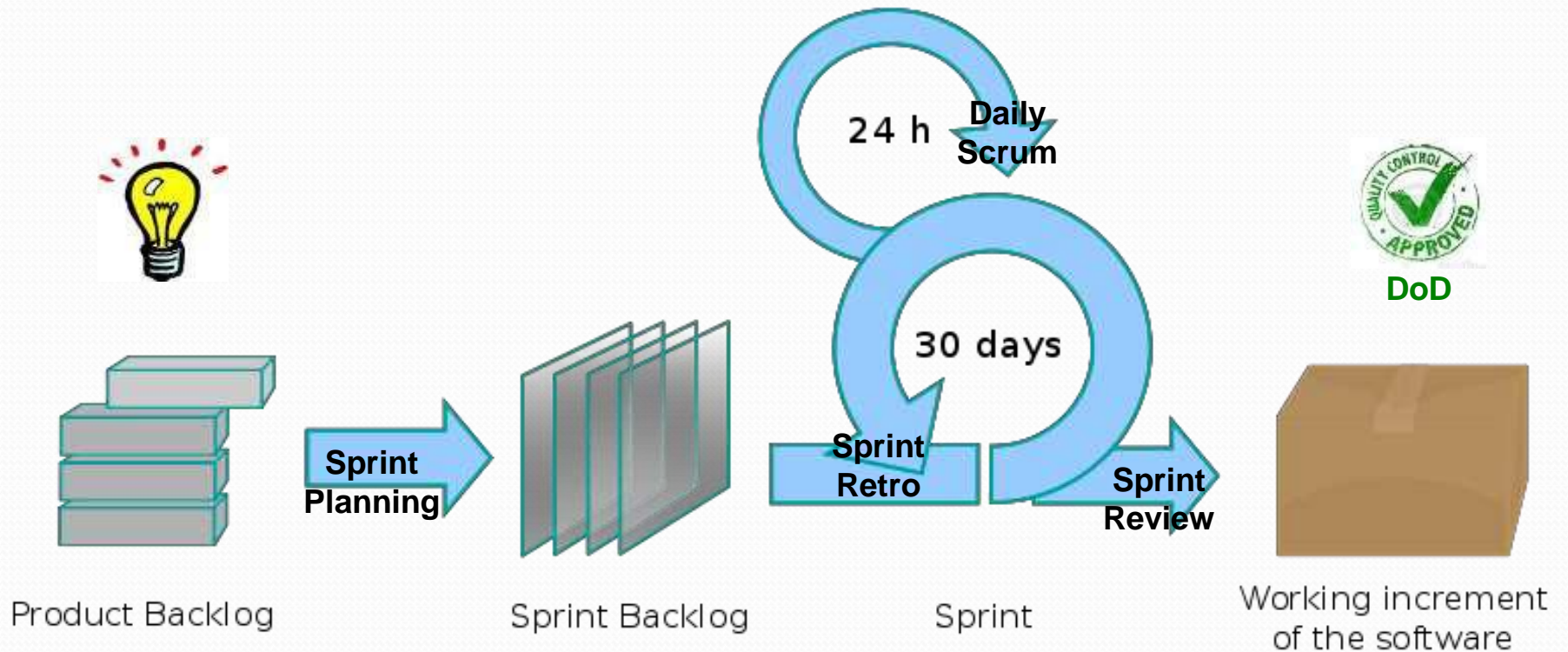


Insert many smaller companies...



ThoughtWorks®

Scrum



Incremental Delivery

Doggone Cards

- ◆ An innovative greeting card company
- ◆ Wants to use modern, agile techniques to develop a new line of merchandise



Doggone Cards

- ◆ We need a new product
 - ◆ But we only do cards
 - ◆ Pen & Ink is our style
- ◆ The concept is simple
 - ◆ Everyone loves a golden retriever, so we'll do a golden retriever card
- ◆ Iterative & Incremental dog drawing



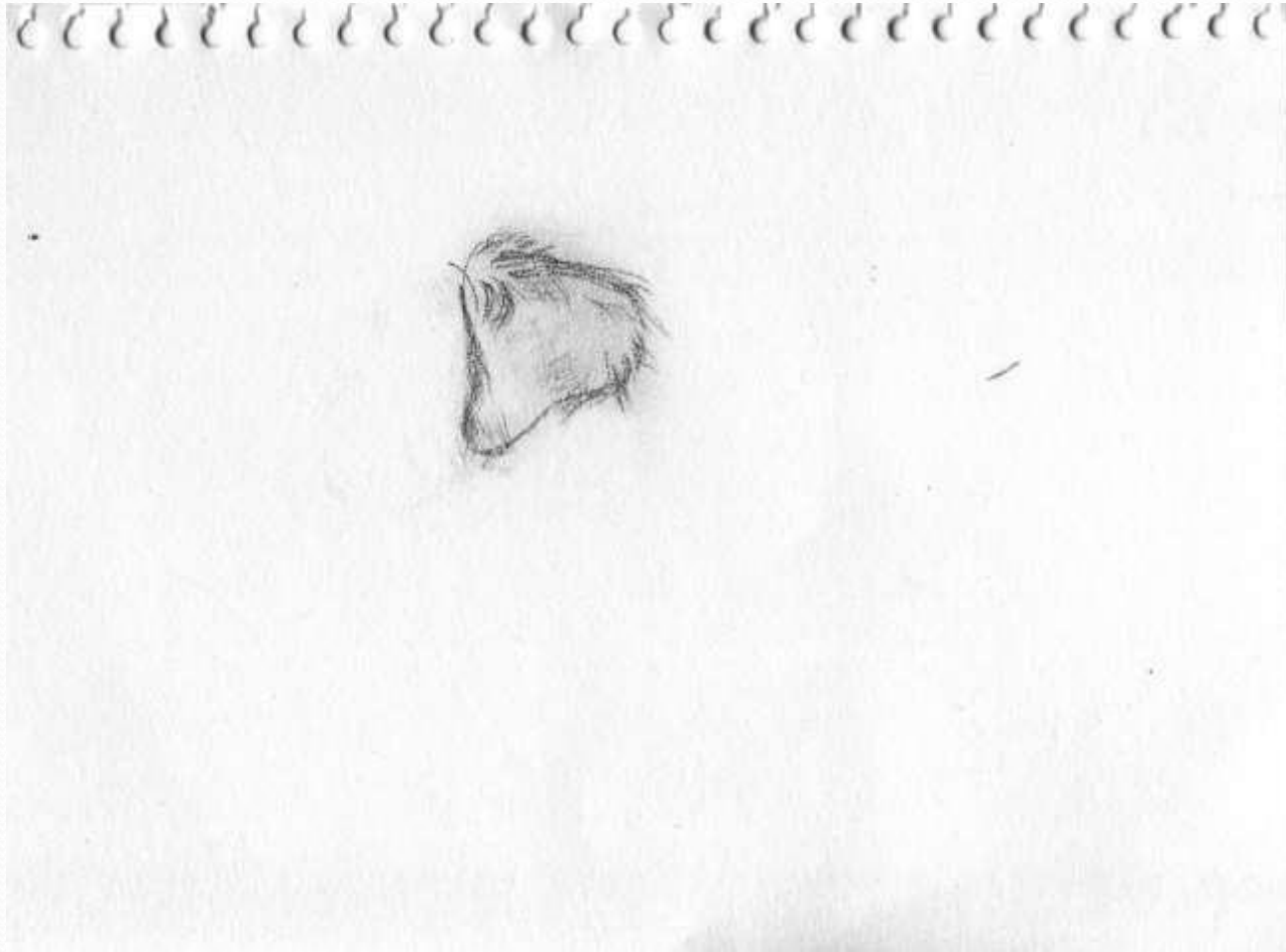
The Concept



Drawing the Dog I



Drawing the Dog II



Drawing the Dog III



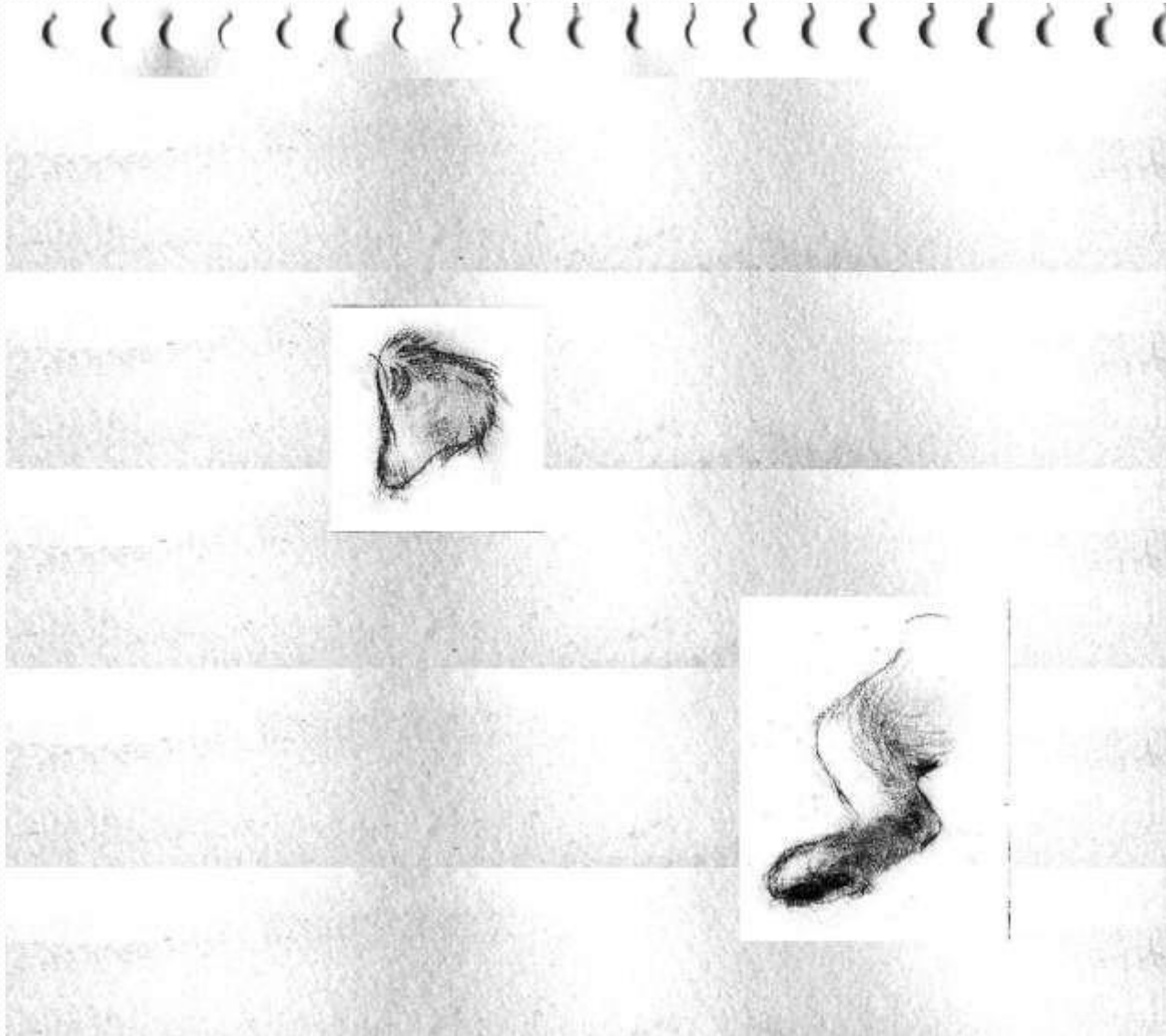


Important Message

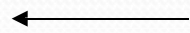
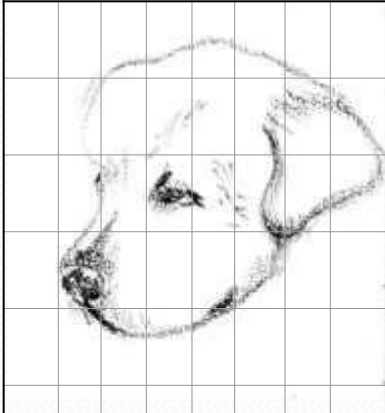
- ◆ We interrupt this golden retriever to bring you an important message from the project manager ...

The Drawing Budget is Exhausted.
Please use the image in its current form ...

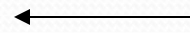
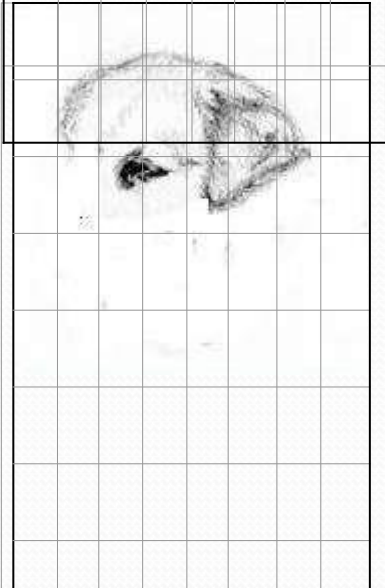
Fit for Delivery?



Learning to Draw



A reasonably good, fit-for-business system delivered to the client can produce revenue



A perfect, but incomplete system that hasn't been delivered produces only cost



Doll House Project

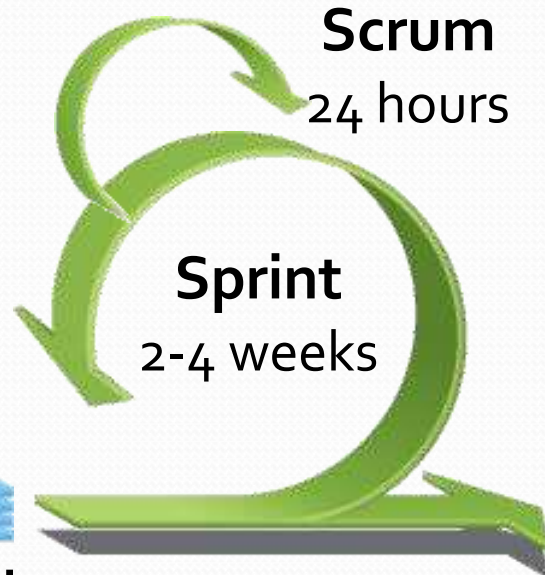
Product Backlog



Sprint Goal

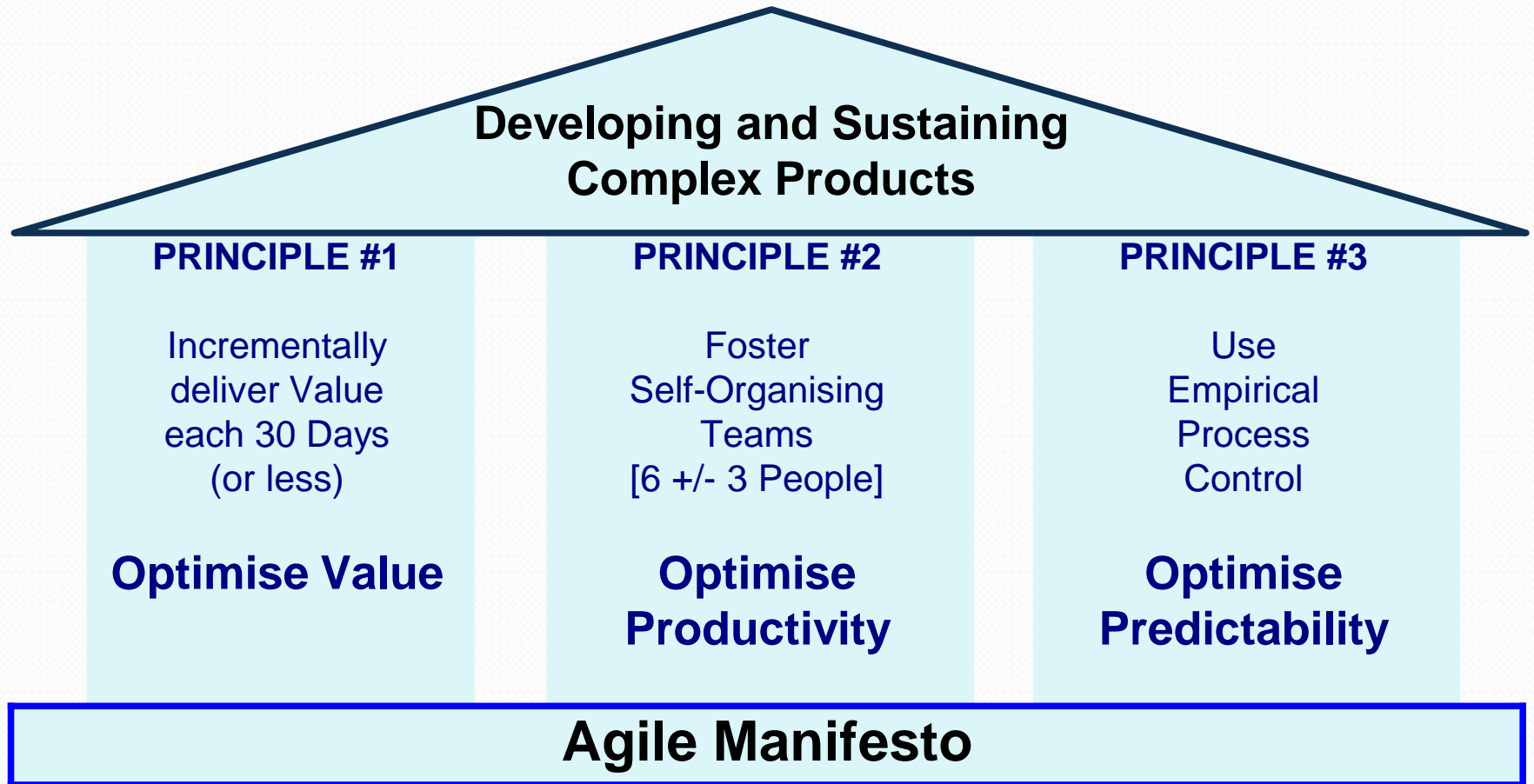


**Sprint
Backlog**



**Potentially Shippable
Product Increment**

Scrum Principles

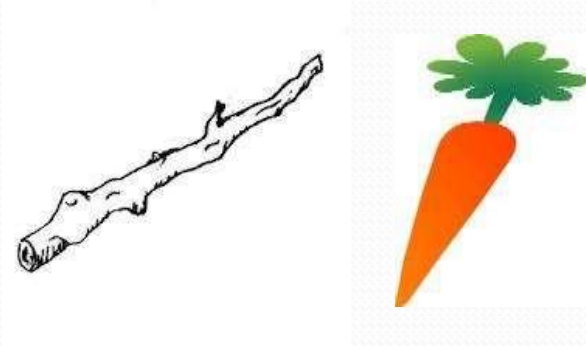


Optimise Value



Optimise Productivity

Extrinsic
Motivation



Simple
Tasks



Intrinsic
Motivation



Knowledge
Work



Self-Organising Teams

Directed Team

Project Management Behaviours

- Prepare detailed staffing plan
- Negotiate for part-time specialists
- Command & control individuals
- Conduct individual performance reviews

Team Behaviours

- Take direction
- Seek individual reward
- Focus on low-level objectives
- Compete
- Comply with processes
- Avoid conflicts

Self-Organising Team

Project Management Behaviours

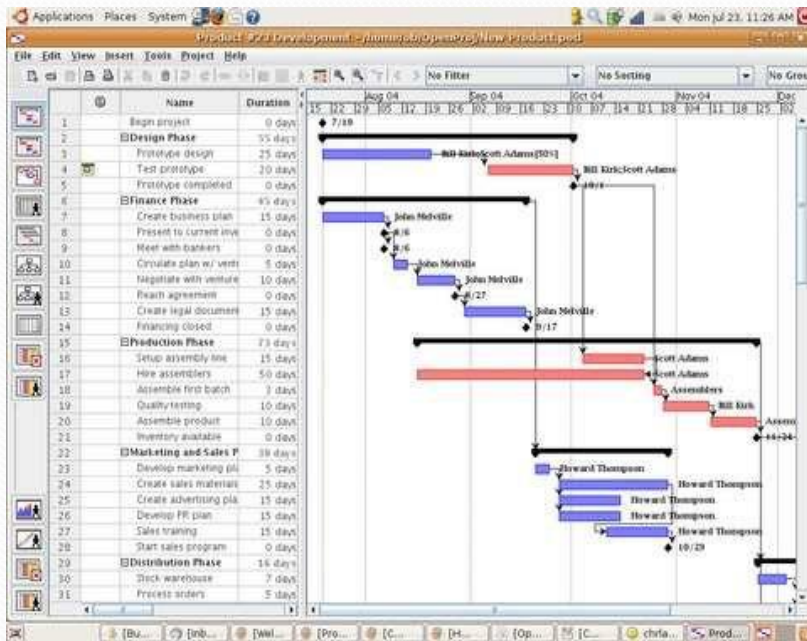
- Gather cross-functional team
- Negotiate for full-time generalists
- Facilitate teams & remove impediments
- Conduct team retrospectives

Team Behaviours

- Take initiative
- Focus on team contributions
- Concentrate on solutions
- Collaborate
- Continuously improve
- Navigate conflicts

Optimise Predictability

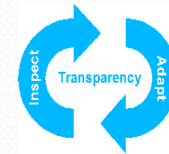
Defined Process Control



Gantt
Chart



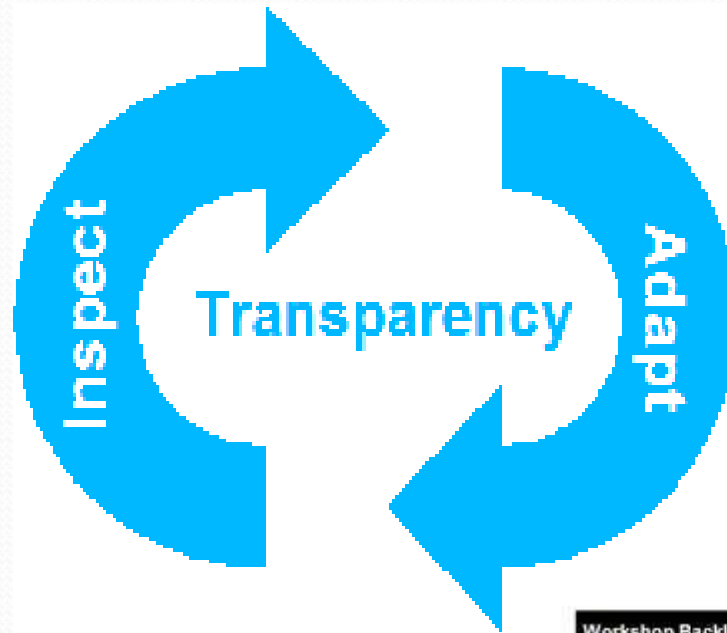
Empirical Process Control



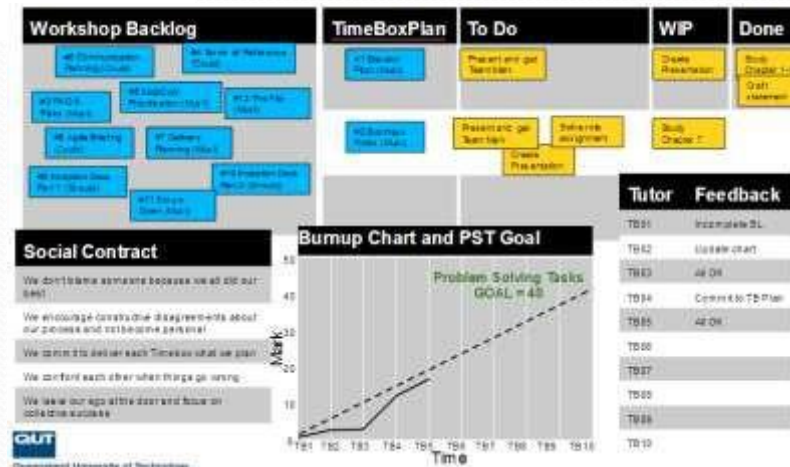
Burndow
n Chart



Empirical Process Control



HOME 47
AWAY 25



Roles



Product Owner

Optimise Value
Product Backlog
Product Increment



Development Team

Self-Organising
Cross-Functional
Sprint Backlog



Scrum Master

Coach & Trainer
Servant Leader
Change Manager

Scrum Roles

Product Owner

Responsible for maximising the value of the Increments delivered by the Development Team

- One person with a vision
- Maximise Return on Investment (ROI) and minimise Total Cost of Ownership (TCO)
- Decide on release date and content
- First point of contact for stakeholders
- Accept or reject work results
- Accountable for managing the Product Backlog
 - Clearly express Product Backlog Items (PBI)
 - Order PBI to best achieve goals
 - Make Product Backlog transparent and understandable to all
- Knowledgeable, empowered and engaged
- Motivates team and celebrates success
- Often full-time role

Development Team

Committed to delivering a potentially releasable Increment of “Done” product at the end of each sprint

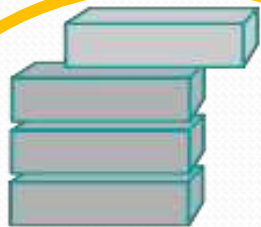
- Typically 6 +/- 3 people
- Self-organising and empowered – only the team estimates PBI's and determines how to turn the Sprint Backlog into an Increment
- Cross-functional – the team collectively possesses all of the skills to create a Potentially Shippable Product Increment
- Shared responsibility – although team members may have specialised skills, responsibility is shared
- Full-time members – exceptions are possible (eg. DB admin)
- No titles – all members are “Developer”
- No sub-teams – no sub-teams for particular domains like business analysis or testing
- Delivers in small chunks
- Builds-in quality

Scrum Master

Responsible for maximising the value of the Increments delivered by the Development Team

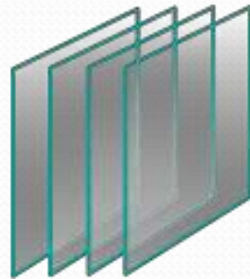
- Coach and trainer for the Product Owner
- Servant Leader for the Development Team:
 - Helps building self organising teams
 - Removes impediments
 - Empowers the Team
- Manager in the Organisation:
 - Causing change to interactions with the Scrum Team to maximise the value created by the Scrum Team
 - Represents management to the project
 - Leading and coaching in Scrum adoption
 - Plans and implements Scrum
 - Works together with other Scrum Masters to increase effectiveness of the application of Scrum in the organisation

Product Backlog

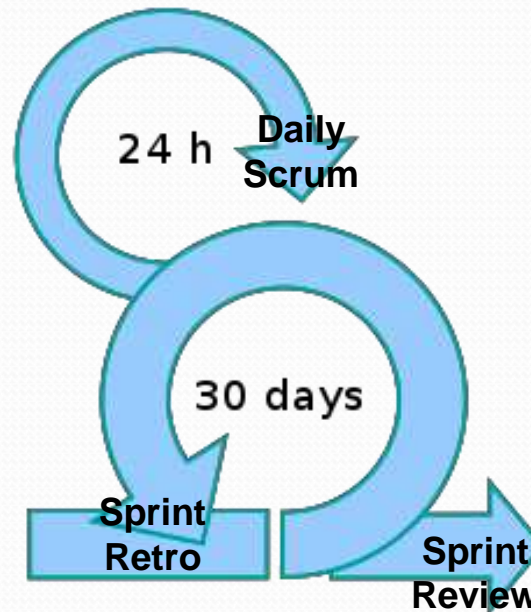


Product Backlog

**Sprint
Planning**



Sprint Backlog



Sprint



DoD



Working increment
of the software

Product Backlog Example

Backlog item	Estimate
Allow a guest to make a reservation.	3
As a guest, I want to cancel a reservation.	5
As a guest, I want to change the dates of a reservation.	3
As a hotel employee, I can run RevPAR reports (revenue-per-available-room).	8
Improve exception handling	8
...	30
...	50

Product Backlog

Detailed
appropriately

Emerging

Estimated

Prioritised
(Ordered)

Product
Owner



Sprint Ceremonies

Sprint: 1-4 Weeks

Sprint
Planning
Part 1:
1-4 Hours

Sprint
Planning
Part 2:
1-4 Hours

Sprint
Review:
1-4 Hours

Sprint
Retro:
1-3 Hours

Daily
Scrum:
15 Mins

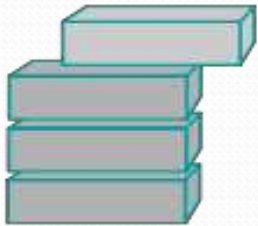
X Week Sprint = X Hour Meeting



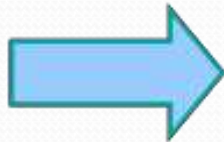
Sprint Planning



**Part One:
What?**



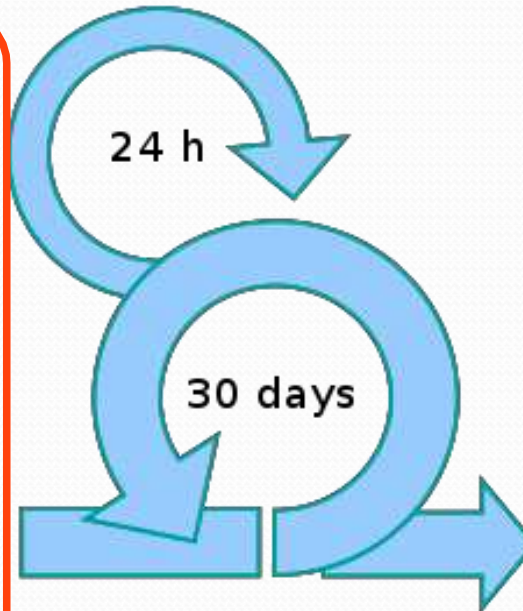
Product Backlog



**Part Two:
How?**



Sprint Backlog



Sprint



DoD



Working increment
of the software

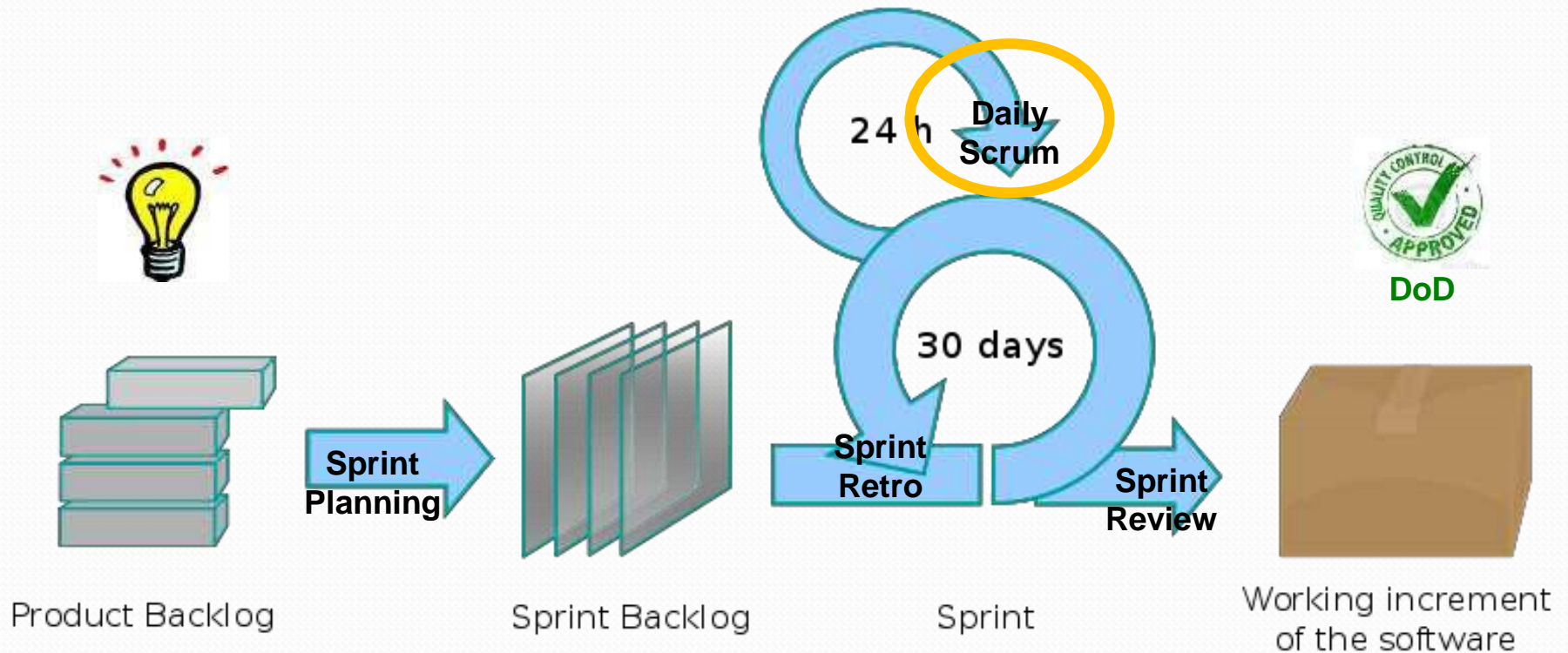
Sprint Backlog (& Goal)



Story	To Do		In Process	To Verify	Done
As a user, I... 8 points	Code the... 9	Test the... 8	Code the... DC 4	Test the... SC 6	Code the... D Test the... SC 8 Test the... SC Test the... SC Test the... SC 6
As a user, I... 5 points	Code the... 8	Test the... 8	Code the... DC 8		Test the... SC Test the... SC Test the... SC 6
	Code the... 4	Code the... 6			

Daily Scrum

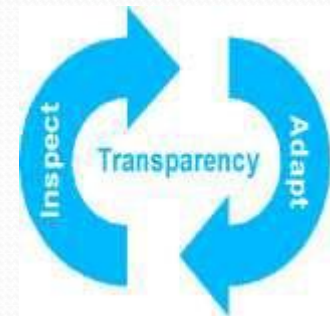
Develop-
ment
Team



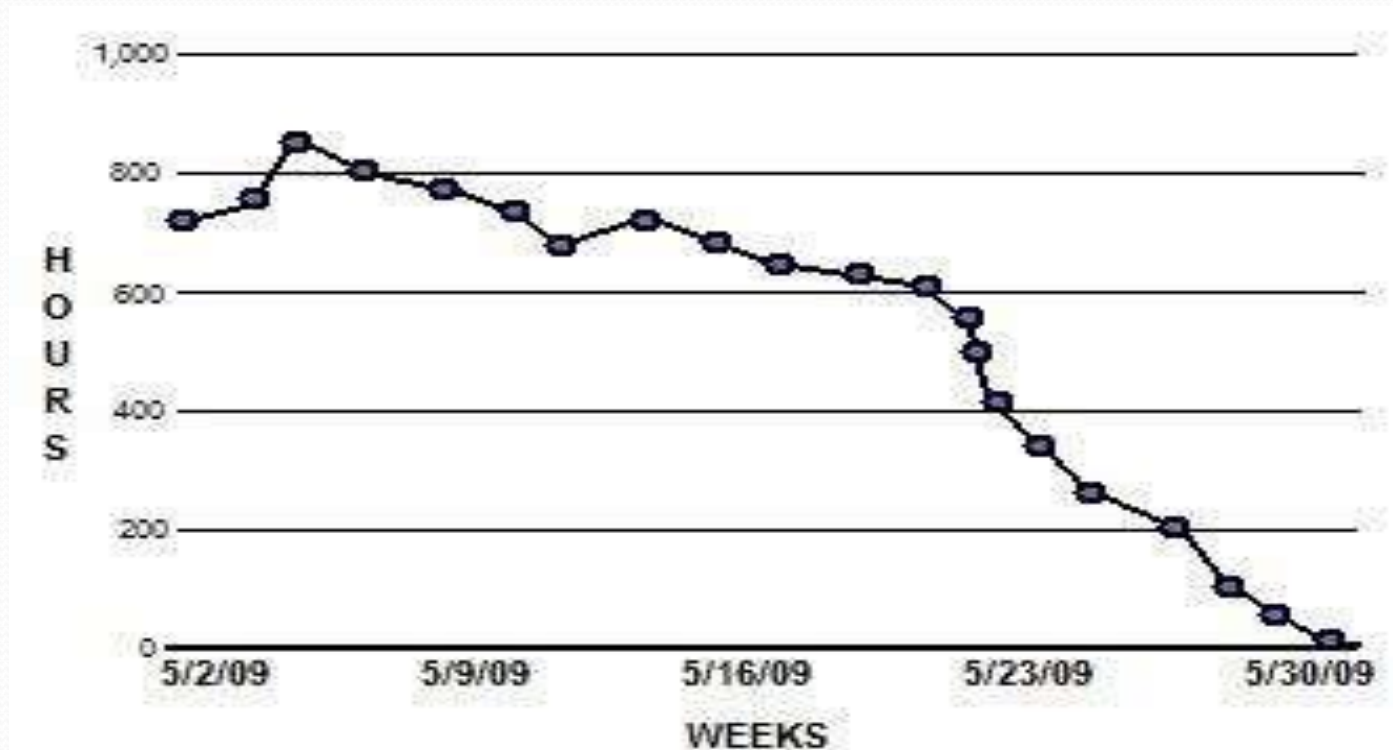
Daily Scrum

Develop-
ment
Team

1. What did I do yesterday that helped the Development Team meet the Sprint Goal?
2. What will I do today to help the Development Team meet the Sprint Goal?
3. Do I see any impediment that prevents me or the Development Team from meeting the Sprint



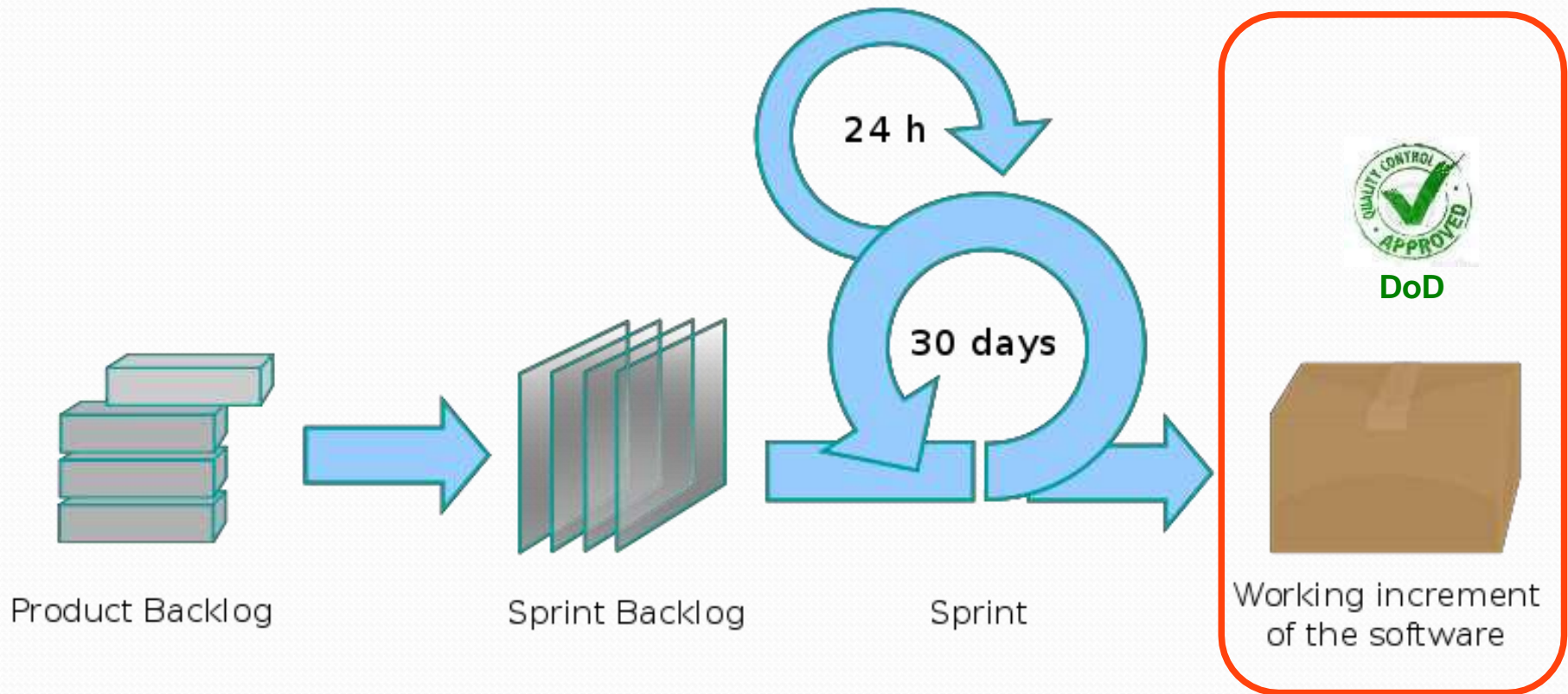
Daily Scrum & Sprint Burn-Down



Development Team

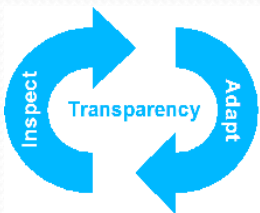
Sprint Review

Product
Owner



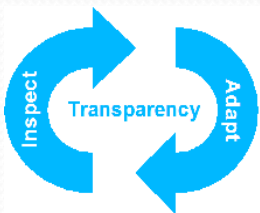
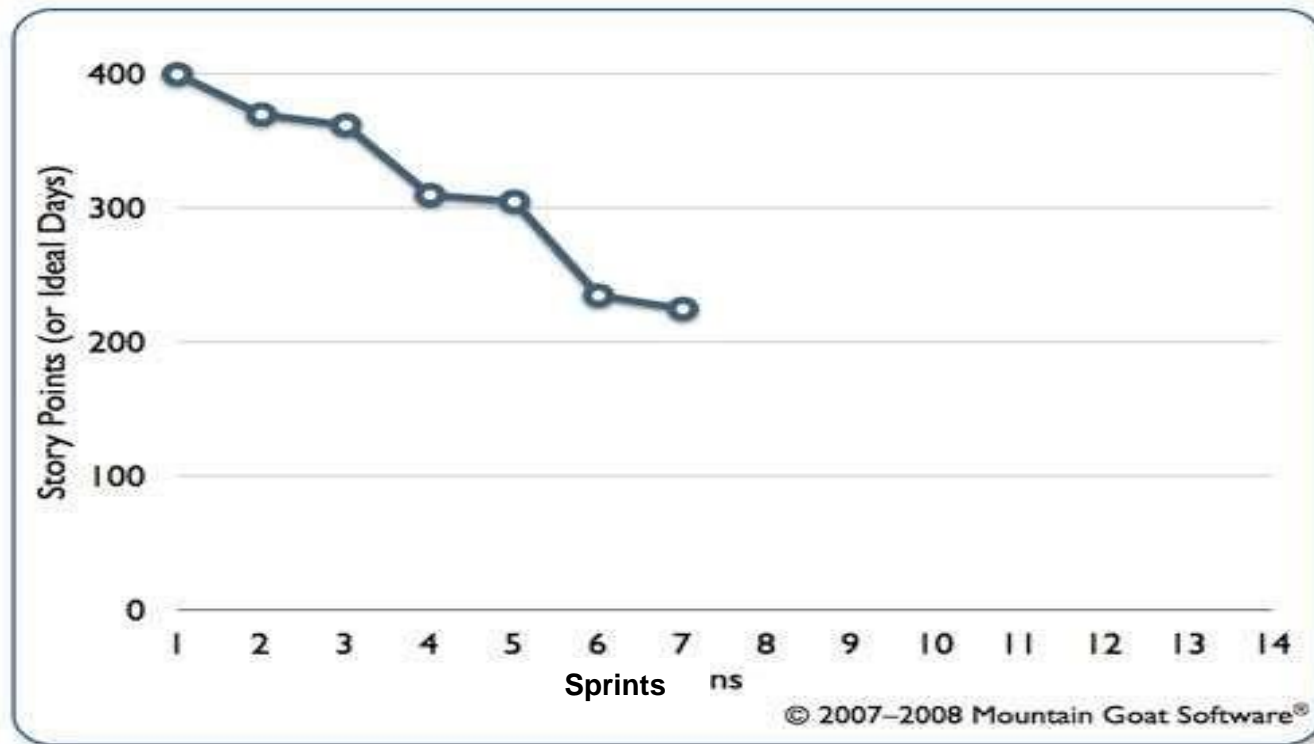
Sprint Review: Product Increment

- ◆ Sum of all completed Product Backlog Items
- ◆ Potentially shippable
 - ◆ Complies with Definition of Done
- ◆ Minimal Marketable Features



**Product
Owner**

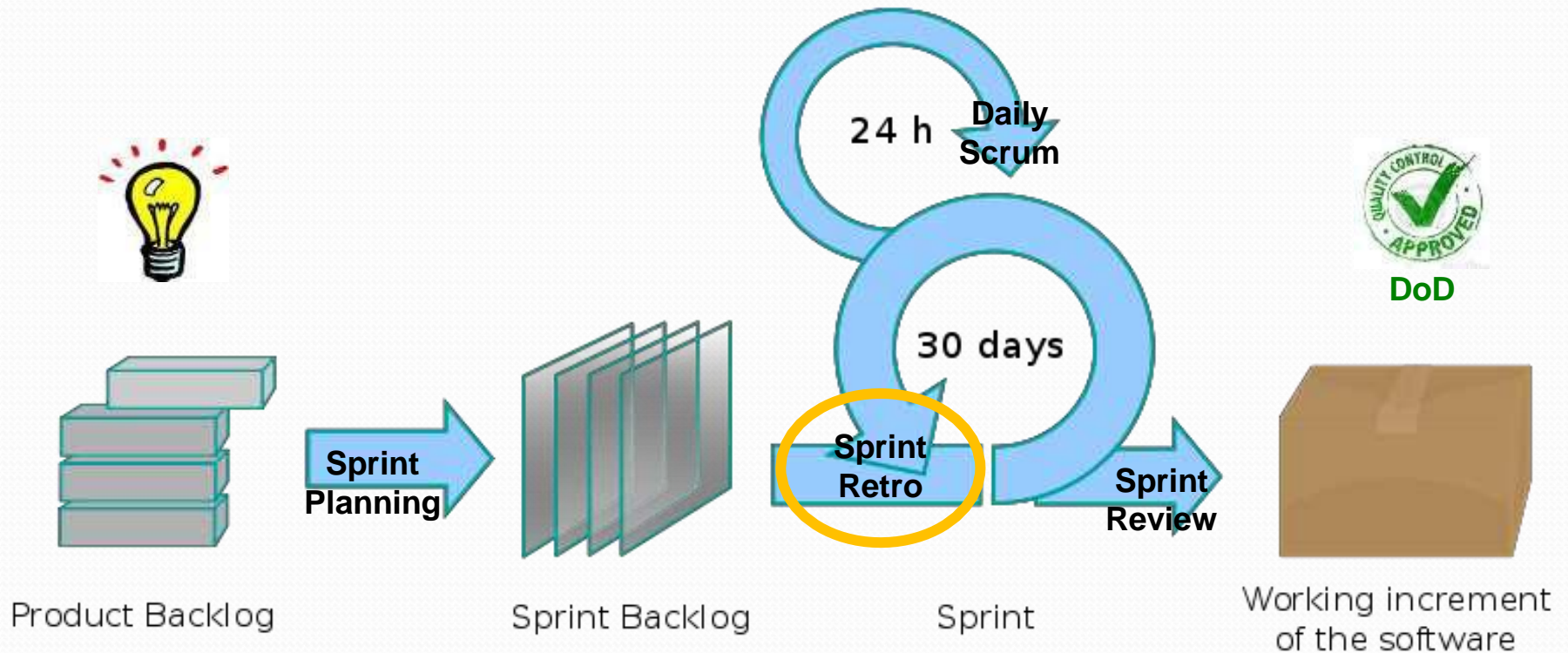
Sprint Review: Release Burn-Down



**Product
Owner**

Sprint Retrospective

Develop-
ment
Team



Sprint Retrospective

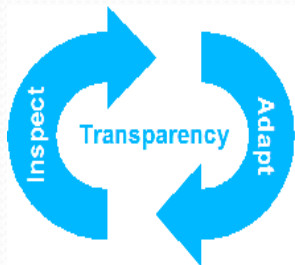
Develop-
ment
Team

Continue
What went well?

Stop
What do differently?

Initiate
Which lessons were
learnt?

Still Puzzling



Prime Directive

Regardless of what we discover, we must understand and truly believe that everyone did the best job he or she could, given what was known at the time, his or her skills and abilities, the resources available, and the situation at hand.

User Stories

What is a User Story?

- **Short description of functionality**
 - textually small
 - short development time (1-3 days)
- **From the user's perspective**
- **Provides value to the user or sponsor**
 - consider both types of clients
- **Must be testable**
- **Provides enough information to make rough estimates**



Why User Story?

- **History of poor requirements capture**
 - large out-dated documents
- **Communication**
 - track requirements
 - cards & BVC
 - up-to-date
 - conversation
 - understand user needs
 - conversation & confirmation

Story Structure

- As a **[role]**, I want to **[do / see / change something]** so that **[outcome]**.

e.g.

- As a **permanent employee** I want to be able to **see my leave balance** so that **I can plan my holidays**.



Story Tools

- Commonly written on index cards
 - then stuck on walls
- Can be managed electronically
 - Jira Agile
 - <https://www.atlassian.com/software/jira>
 - <https://sefteach01.qut.edu.au/>

Story Cards

Front of Card

173

As a student I want to purchase a parking pass so that I can drive to school.

Priority: ~~High~~ Should Estimate: 4

Back of Card

Confirmations:

~~The student must pay The correct amount~~
One pass for one month is issued at a time
The student will not receive a pass if the payment isn't sufficient
The person buying the pass must be a currently enrolled student.
The student may only buy one pass per month.

Story Wall



Story Example

Story Ref #	Feature	Story Title	As a	I want to	so that
1	Upload Media	Upload Audio File	Uploader	upload an audio file	it is safely stored in a universally accessible location

Questions	Comments	Business Priority	Story Point Estimate	Status	Release	Iteration
Who can access it once it is uploaded?	<i>Safely</i> assumes cloud storage mechanism provides redundancy	High	2	Story written	1	1

The Whole Story

- ***Card*** – initial written description
 - often on index cards
- ***Conversation*** – between developers and customer representatives
 - customer driven design
- ***Confirmation*** – tests to determine when implementation complete
 - initial criteria written on story card
 - full tests in automated test suite



Conversations

- **Details are discovered by talking with the customer representatives.**
- **Conversations occur whenever someone needs information.**
 - not “once off”
- **Story cards are a starting point for a conversation.**
 - they don’t record the requirements
 - may record notes on card during conversation
- **The UATs become the requirements.**
 - we’re done when we pass the tests



INVEST in your Stories

- **Independent**
 - dependencies make planning, prioritisation and estimation difficult
- **Negotiable**
 - details are worked out in conversation
 - between developers and customers
 - too much detail limits the conversation and options
 - too easy to think all detail is in story
- **Valuable**
 - must provide value to customer
 - get customer to write stories



INVEST in your Stories

- **Estimable**
 - at least to start with ballpark estimates
 - prioritisation and planning depends on this
 - problems: lack of domain knowledge or story too big
- **Small**
 - representing a few days in person effort
 - the smaller the stories, the more accurate the estimates
- **Testable**
 - need completion criteria
 - we don't develop what we can't test



Example Stories

- As a *marketer* I want *the Digital Workspace to look like a QUT site* so that *we project a consistent image*.
- As an *IT support* I want to be able to *post an outage / downtime notification* so that *I can inform people that the portal is out of service*.
- As a *web developer* I want to be able to *capture usage data in XML log files* so that *we can analyse patterns of usage*.

Next Week Lecture – Story Time

- ◆ Stakeholders – the client
- ◆ Scope – how much
- ◆ Specifications – what they want
- ◆ You'll need index cards for workshop 2



173. Students can purchase parking passes.

Priority: 8

Estimate: 4

Next Week Workshop – Story Time

- ◆ Requirements Analysis
- ◆ Development of User Stories
 - ◆ Email your developed user stores to client team at least two day prior to your next workshop