agile tour 2011

Agile & Scrum

Everything you wanted to know, but were afraid to ask

goshido

gerard.hartnett@goshido.com

Key Patterns

Agile can be applied in any kind of business

Bottom Up Planning (take care of days...)

Simple Top Down Plan - it will change

Adaptable more important than predictable

Feedback Loops - OODA & PDCA

Subtle Control, Encourage Self-Organisation

Overview

Introduction

Patterns, PLOPS and OOPSLA

Scrum

eXtreme Programming

Conclusion

Introduction

23 years SW experience, 13+ Agile

Currently: Goshido, Limerick

Intel, Basis & Tellabs, Shannon

Digital, Galway

Motorola, Cork, & USA

QC-Data, Cork & Canada

Patterns, PLOPS and OOPSLA

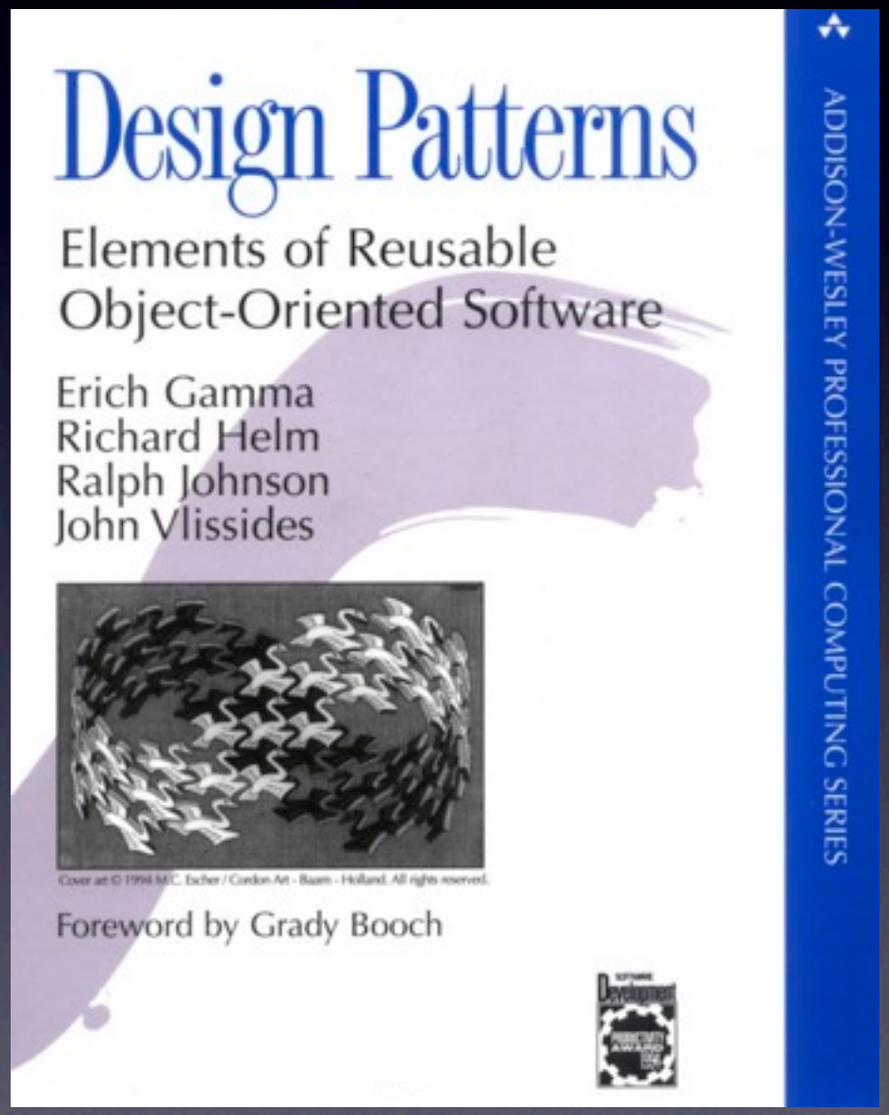
Patterns

Reusable solution to a common problem

Borrowed from Architecture (Alexander)

1987 OOPSLA, Beck & Cunningham

1994, Gang of Four

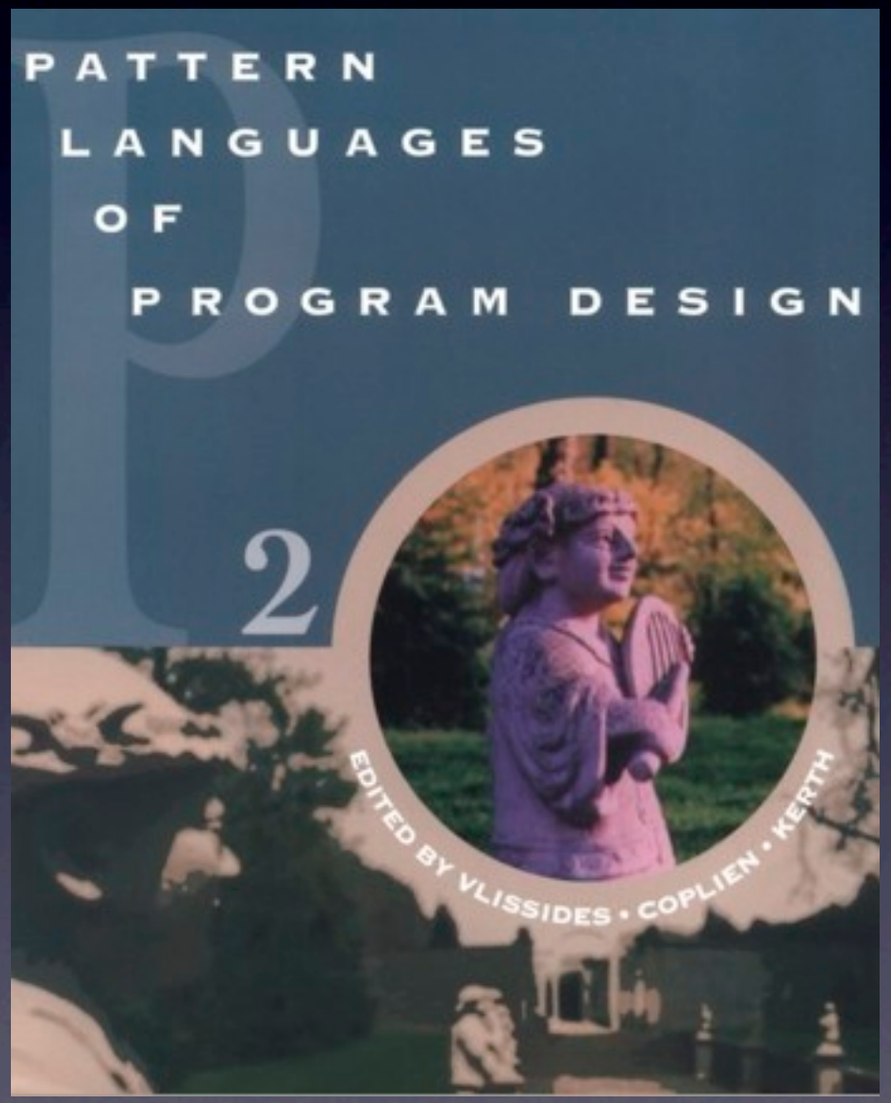


PLOPs

Pattern Languages of Programs

Conferences Started 1994

PLOP-2, paper on Episodes (Cunningham)



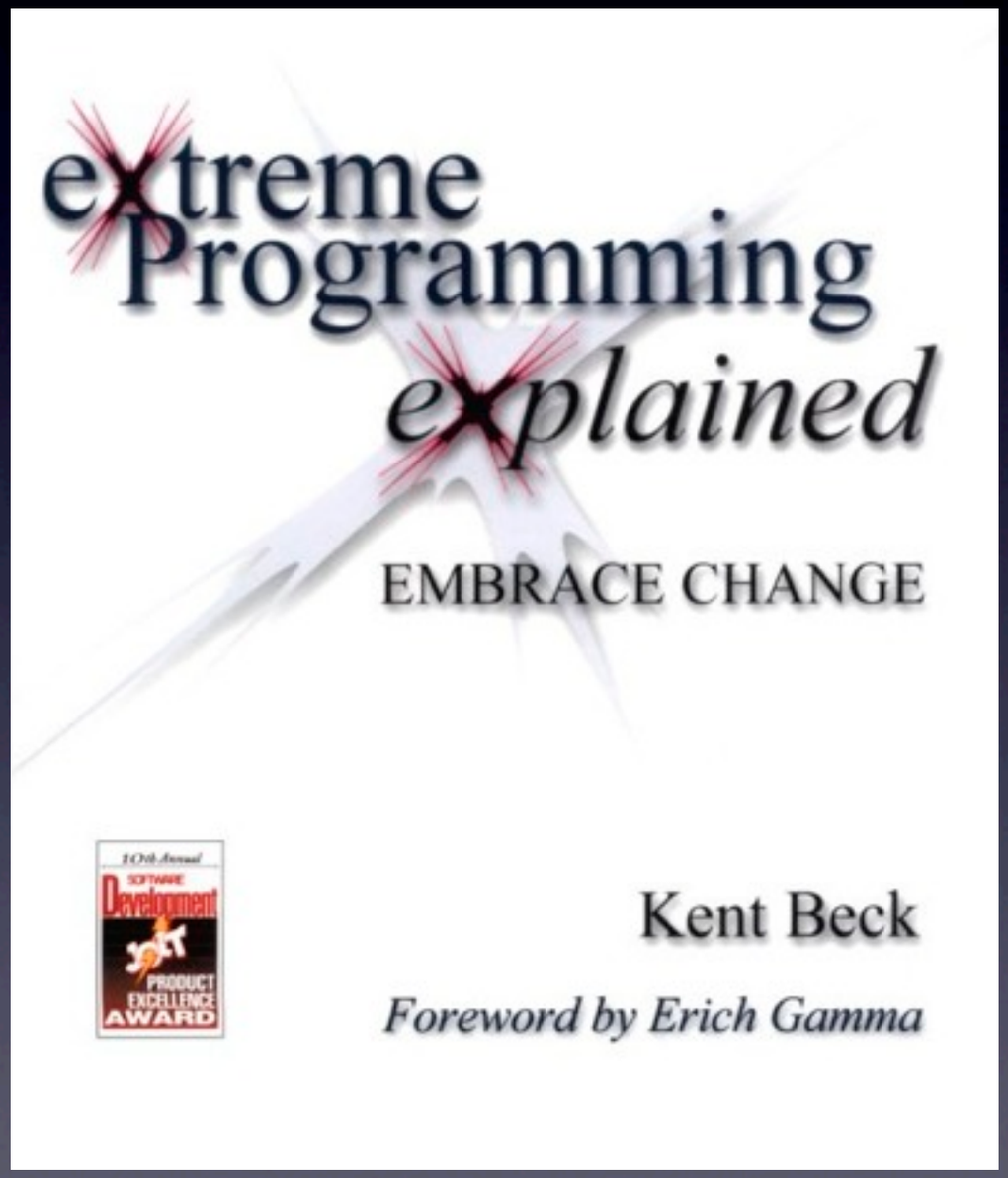
OOPSLA 98

Birds-of-a-feather session

100s arrived

One page handout of the key practices

Diagram of how they related to each other



eXtreme Programming

Overview

A light-weight methodology

For small-to-medium-sized teams

Vague or rapidly-changing requirements

Values: communication, feedback, simplicity, courage and respect

Practices

Quick Plan

Small Releases

Metaphor

Simple Design

Testing

Refactoring

Pair Programming

Collective Ownership

Continuous Integration

40-Hour Week

On-Site Customers

Coding Standards

Pair Programming

Photo credit flickr.com/people/lucadex

Pair Programming our Experience

The most controversial practice

Some components pair programmed

Lowest defect rates

Smaller code sizes

Small sample set

Scrum

Specific Projects

1998 Team of 3 in Tellabs

2000 Team of 5 in Intel 5.5 month project

2001-02 Intel team of 33, 15 month project

Many smaller projects

2008-11 Goshido team of 3-7

New New Product Development Game

Article in Harvard Business Review 1986

Study of companies in Japan & US

Fuji-Xerox, Canon, Honda, NEC, HP, 3M


HBR

JANUARY–FEBRUARY 1986

The New New Product Development Game

Hirotaka Takeuchi and Ikujiro Nonaka

Six Characteristics

Built in instability - broad challenging goal

Self-organising teams

Overlapping Development Phases

Multilearning

Subtle Control

Organisational transfer of learning

Scrum

- Computer guys borrowed from "New New..."
- Called it Scrum
- I suspect they were never in a real Scrum

Photo credit flickr.com/photos/bohane/

Scrum Overview

You can manage any project with Scrum

I've used on tiny and large projects

Not rocket science or expensive (although an "industry" has popped up)

It works & viral

“Opportunities”

All projects have chickens and pigs

Pigs are happy: no status reports or boring meetings

Chickens are happy: they know what’s happening without getting in the way

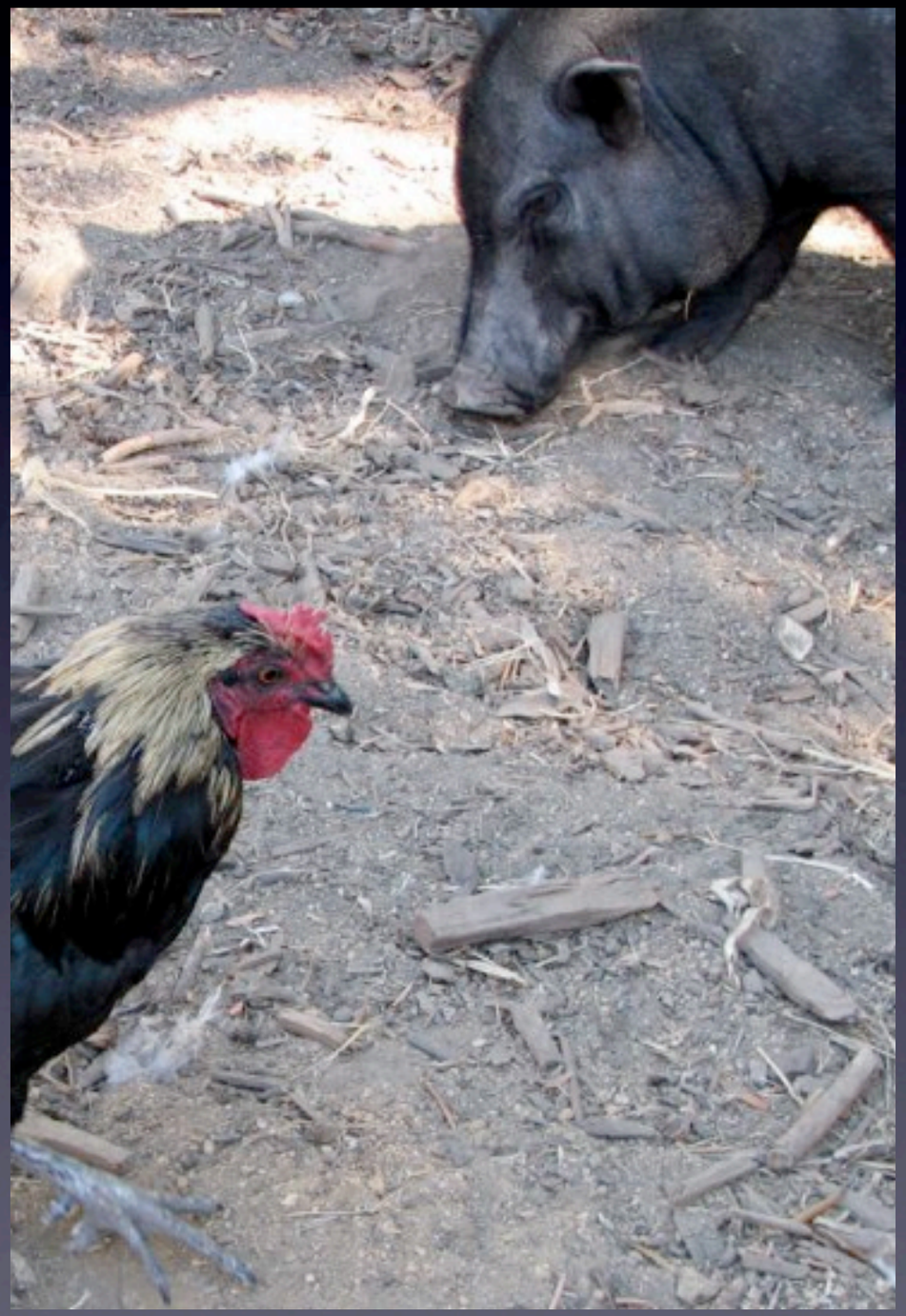


Photo credit flickr.com/photos/7326810@N08

The Theory

Developing SW is a complex adaptive system

Traditional planning - illusion of control

Empirical process not a “defined process”

Complexity theory - better to do frequent samples and small adjustments

Adaptability is better than predicability

The Team

Should be small (6 max) - 2 pizza rule

If bigger, split

Ideally co-located (can work distributed)

The Project

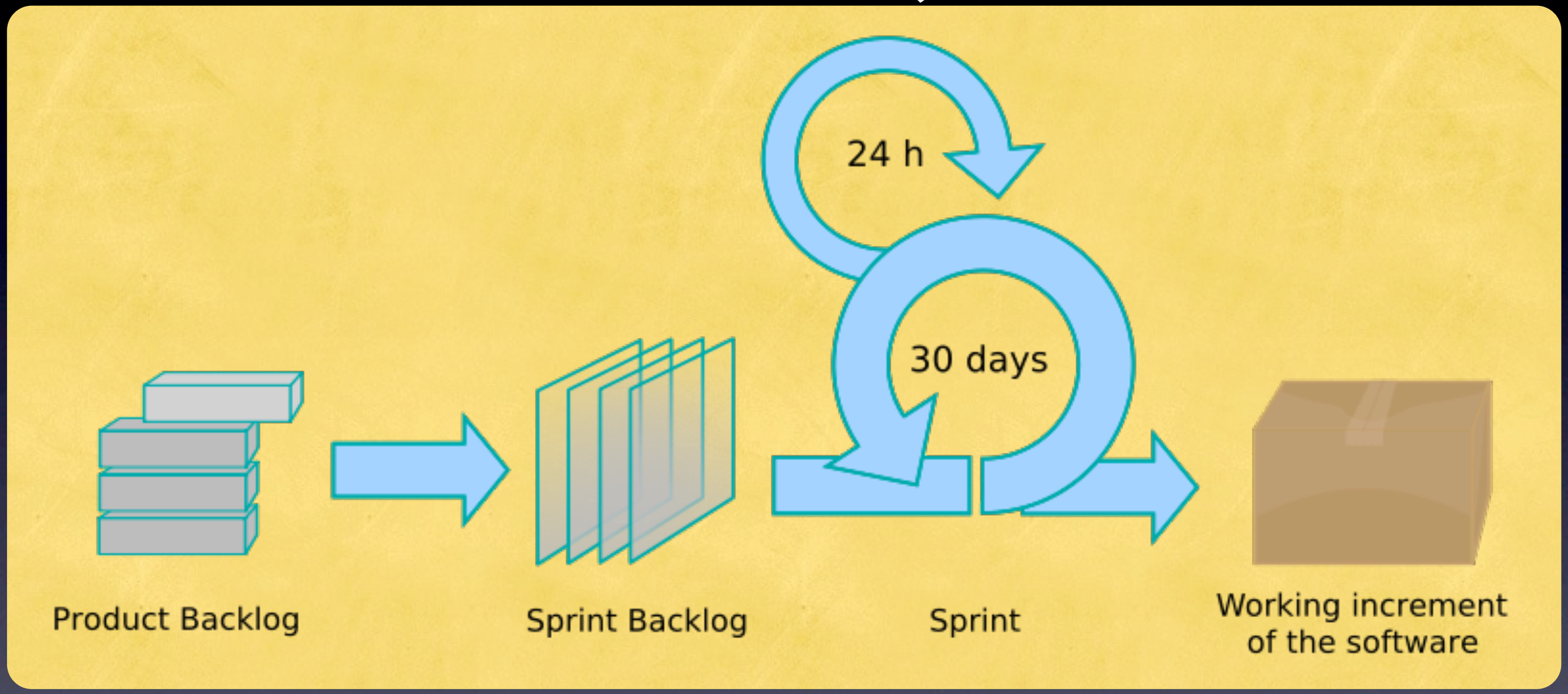


Diagram by Lakeworks http://commons.wikimedia.org/wiki/Image:Scrum_process.svg

- Punctuated Equilibrium - balance team focus, management visibility, and adaptability

The Sprint

Easier for team to focus on something a month away

Daily short meetings

At the end - write-up or demo

The Daily Meeting

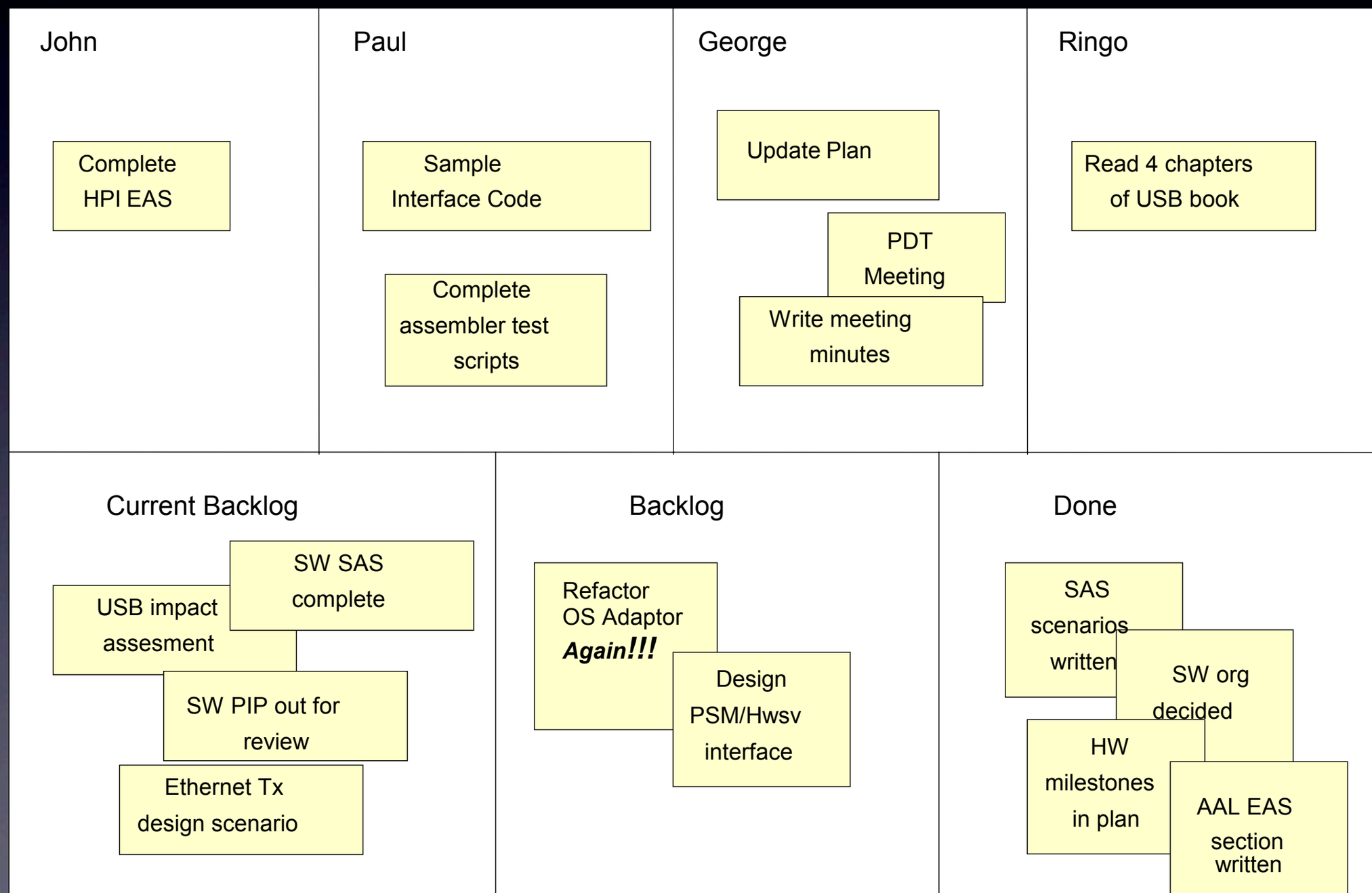
KISS - 15 minutes max, same time & place

3 questions

- what did you do? (binary - all or nothing)
- what roadblocks?
- what will you do by tomorrow?

Everyone responsible for their own tasks

Loads of Post-its stuck to a big sheet of paper



Consequence of "Loads of post-its"

Low effort - visualisation of progress

Preparation is fast - post-its on monitors

Occasional trips to the "big sheet"

Not great for distributed teams

3M make loads of money

Planning

More KISS

No complex Gannt chart

Plan is a series of sprints

Dependencies only at a sprint level

Can use sophisticated estimation techniques

Shorter sprints at the start and end

Planning - Full Project

effective number of engineers		2.6	2.6	2.7	2.8	2.8	2.8	
Sprint Tasks	Baseline	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Unassigne
Atmd.HLD.workshops	12.0							12.0
Aal5/Aal0/Aal2.workshops	2.0							2.0
Fpath	64.5							64.5
QMgr	43.5							43.5
Atmd	111.5							111.5
Aal5Acc.FuncSpec.Draft/Review	10.0							10.0
Aal5Acc.DesignSpec	7.0							7.0
Aal5Acc.Code	6.0							6.0
Aal5Codelet.Spec	7.5							7.5
Aal5Codelet.Code	7.0							7.0
Total	271	0	0	0	0	0	0	271.0
Fixed Overhead								
Holidays.Public	4.0							4.0
Holidays.Vacation	45.0							45.0
Training	4.0							4.0
Total	53	0	0	0	0	0	0	53.0
		0	0	0	0	0	0	

Planning - Sprint

At the start check project plan & look at new backlog items

Allocate to individuals - could use simple spreadsheet

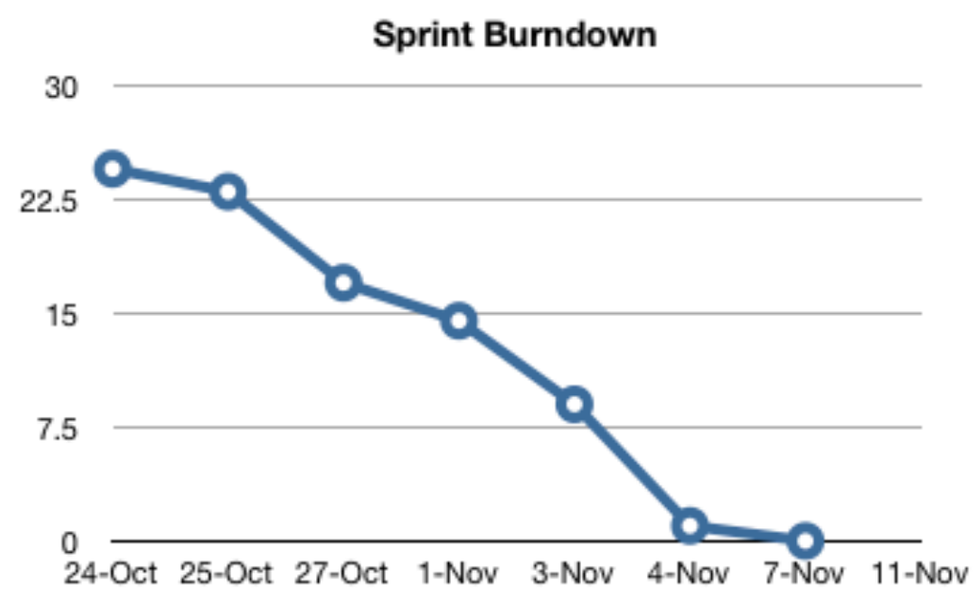
Effort in “ideal days” or “story points”

Inside/Outside end-date

At the end: write-up, demo, release

Sprint - Sheet

	John	Paul	George	Ringo	
Availability	50%	80%	80%	80%	
User authentication	4	4			
Password crypt			1.5		
C				3	
D			3		
E				4	
F		3	2		24.5
	8	8.75	8.125	8.75	
Max	8.75				
Sandbag	12%				
Duration left	9.8				
24-Oct	24.5				
25-Oct	23				
27-Oct	17				
1-Nov	14.5				
3-Nov	9				
4-Nov	1				
7-Nov	0				
11-Nov					



The Big Chickens

Start to trust it - less reporting needed

Can show up at daily meetings - must remember they are chickens

Can add new items to backlog

Other possibilities - earned value, wide-band-delphi, comparison to estimates

Scrum Conclusion

Simple, low-overhead, adaptable any kind/ size project

Everyone gets involved in planning and tracking, team “gells”

The chickens can actually contribute

Scrum was more sticky than eXP

Cloud platform that helps teams

- **Focus** on doing the work that matters
- **Clarify** communication
- **Scale** 400 people on 1 project or 4 on 100

Blend of Agile, Workflow & Attention Mgt

Training & Consulting for large clients

Conclusion

Key Patterns

Agile can be applied in any kind of business

Bottom Up Planning (take care of days...)

Simple Top Down Plan - it will change

Adaptable more important than predictable

Feedback Loops - OODA & PDCA

Subtle Control, Encourage Self-Organisation

Next Steps - 72 hours

- ☐ Read about Scrum/XP on wikipedia
- ☐ Read the paper: Fitzgerald, Hartnett "Agile inside Intel" (goshido.com/_private/Agile-Inside.pdf)
- ☐ Goshido blog "Agile for any organization - a guide"
- ☐ New management books (Denning, Hamel, Appelo)
- ☐ Consider trying Scrum with your team

Thank You