

Strategic Application of Software Development Process for Business Oriented Projects

Glenn Engstrand
President and Founder
Dynamical Software, Inc.



Project Failure Studies

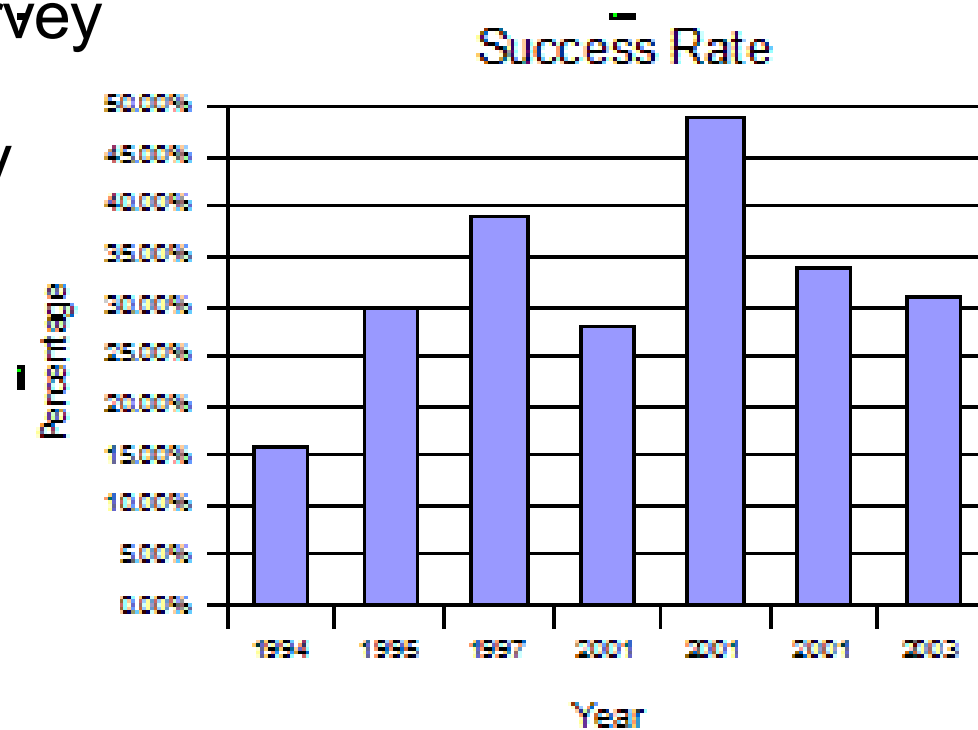
Robbins-Gioia Survey

Conference Board Survey

KPMG Canada Survey

CHAOS Report

OASIG Survey



ACM Queue

1 Minute Risk Assessment Tool

Use of Inappropriate Methodology

Lack of Customer Involvement

Lack of Formal Project Management Practices

Dissimilarity to Previous Projects

Project Complexity

Requirements Volatility



IEEE Software

Frequently Forgotten Fundamental Facts About Software Engineering

Requirements Churn

Overly Optimistic Estimates

Rush to make deadlines increases technical debt

Most Significant Phase is Feature Enhancement Maintenance

Quality Control

Peer Review

White box testing

Why Projects Fail

Information Overload

A new methodology is overly hyped resulting in high expectations with inappropriate and misplaced enthusiasm.

Just Do It

An obsessive focus on meeting short term milestones and budgetary goals results in excessive technical debt resulting in rigidity over time.

Common Theme

Lack of understanding as to how to apply fundamentally sound academic theory to practice in a business context.

The only thing that the customer is willing to directly pay for is code.

Information Overload

Theory

Popular

Over hyped

Complex

Initial Enthusiasm

Apply Everything

Nerds Competing

Eventual Disappointment

Too much overhead

Burn Out

Discard Everything



Examples

Computer Aided Software Engineering

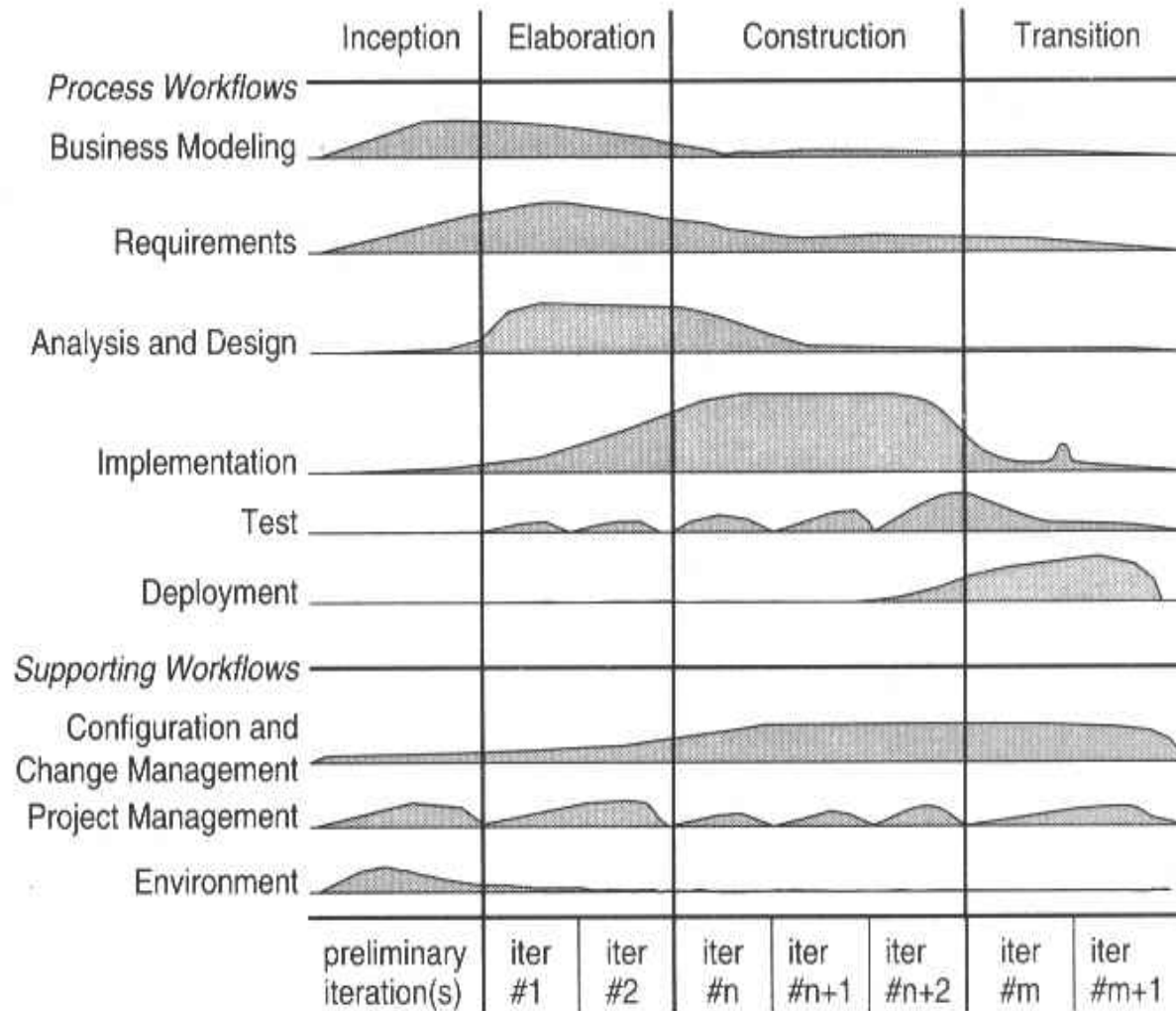
Rational Unified Process

eXtreme Programming

Agile Methodology

SCRUM

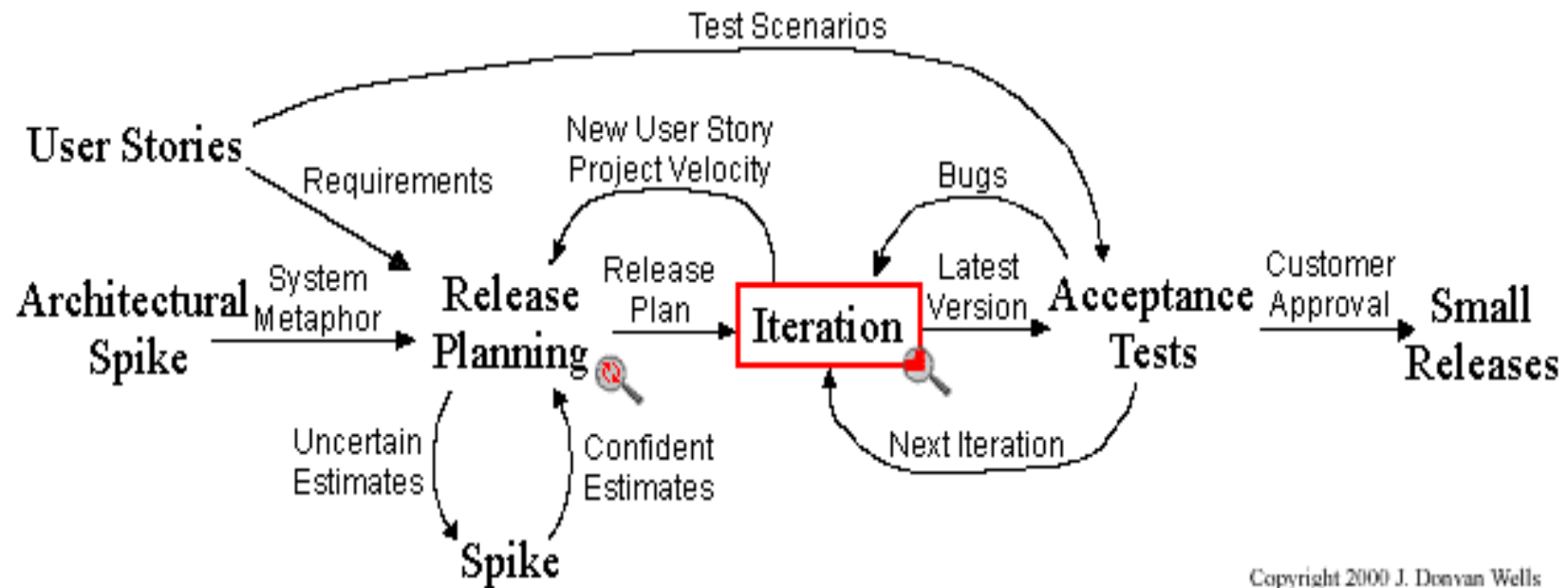
Rational Unified Process



eXtreme Programming



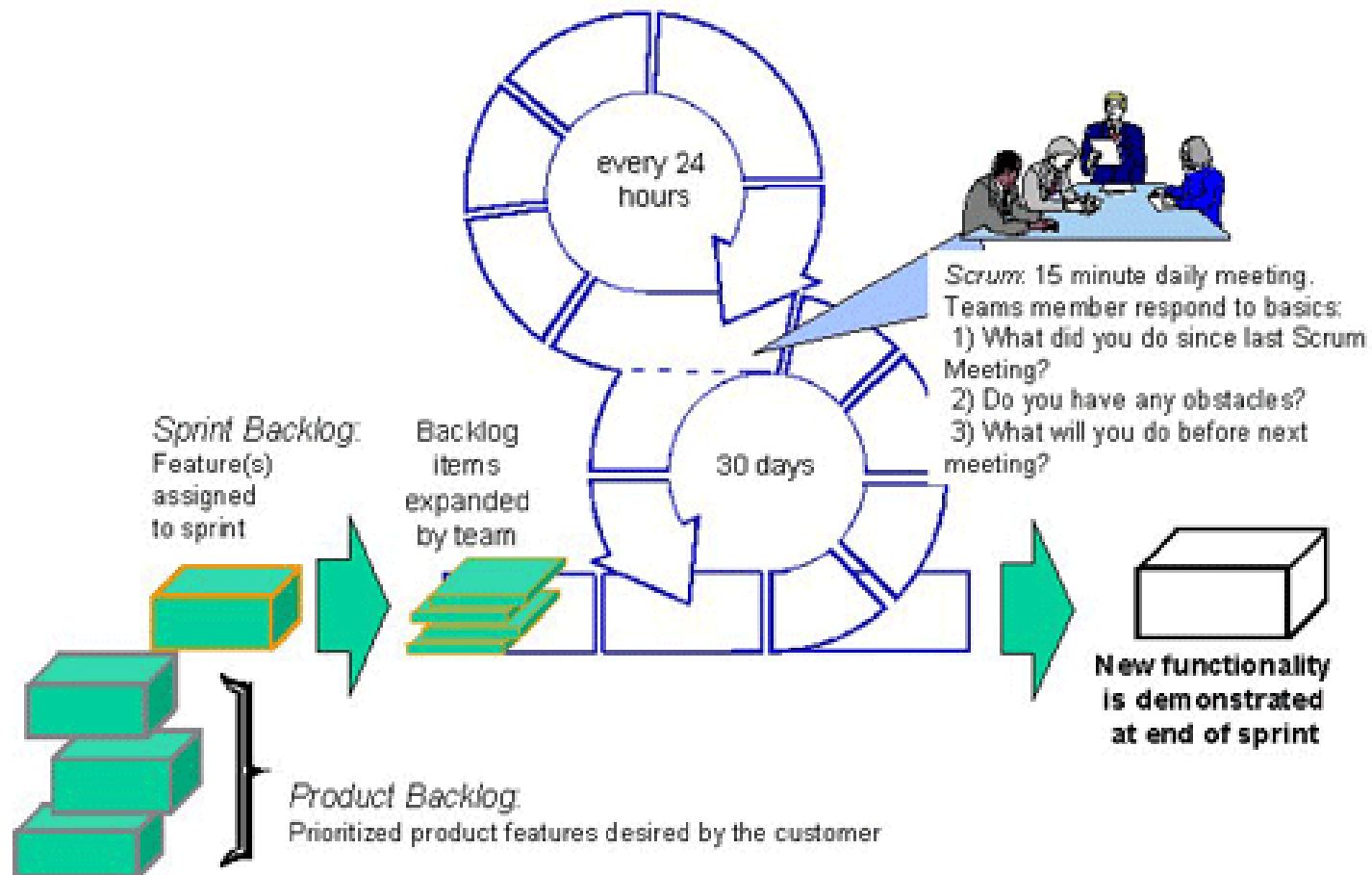
Extreme Programming Project



Agile

- customer involvement
- embrace change
- short iteration cycles
- no silos
- motivation
- face time
- working software
- sustainable
- peer reviewed design
- simplicity
- self organizing
- optimizing

SCRUM



Solution

Cafeteria Approach

Adopt only what makes sense

Don't try to apply everything

Don't change corporate process just to learn something

Expectation Management

How Adoption Affects Productivity

Initially Lower

Eventually Higher



Just Do It

The Pendulum Swings

See Previous Scenario



Penny Wise and Pound Foolish

Developers

Under Educated

Inexperienced

Cheap

Managers

Missed Milestones

Time Micro Management

Short Term Thinking

Solution

Focus on the Total Cost of Ownership

What makes some bugs expensive and others cheap?

The earlier in the SDLC phase

The longer it takes to fix

The higher the cost of the bug

Technical Debt



Capability Maturity

- Capability Maturity Model
- Software Engineering Institute
- Carnegie Mellon University
- Originally a USAF funded study
- Managing the Software Process
- Not just for software
- Now called CMMI (Integration)

Capability Maturity

Initial

heroics

Managed

improve quality

Defined

formal process

Quantitative

taking measurements

Optimizing

track and improve

Look For Pain Points

Initial

burn out

Managed

high bug counts

Defined

reign in the rogues

Quantitative

your work is done
here

Optimizing

Advocating Process

- Introducing process into a business setting is a sales job.
- Expect resistance because of the perception that all that the customer is buying is code.
- You must get buy-in from management.
- You must also get buy-in from engineering.

Selling to Management

Deliverables

user stories, walk-throughs,
story boards

use cases

UML

FPA

gant charts

burn-down charts

Talking Points

gain deep insight and
understanding of your market

prioritizes features

remove mistakes early, mentoring,
reduce technical debt

formalized approach to
estimates

time and resource management

know where you are in the
project

Selling to Management

Methodology

information architecture

eXtreme Programming

Agile

SCRUM

Talking Points

customer relevance

good for when you have junior coders

keeps the team focused and
learning what the customer wants

increases team motivation

Selling to Engineering

Deliverables

user stories, walk-throughs,
story boards

use cases

UML

FPA

gant charts

burn-down charts

Talking Points

assess requirements quality
before committing to estimates

technical insight to requirements

software quality

takes estimating off your plate

fewer distractions from
management

permitted to revise estimates

Selling to Engineering

Methodology

information architecture

eXtreme Programming

Agile

SCRUM

Talking Points

coders hate to come up with labels

coders hate to document

get to complete on your promises

flattering to be served by
management

Code Roller™

- collaborative software development project life cycle management solution
- community where entrepreneurs and engineers get together to produce great software
- raise the level of intelligent awareness of business savvy SDLC process
- free to use



Code Roller™

- Collaborative
- Life cycle
- Intelligent
- Cybernetic
- Knowledge based
- Software
- Product
- Project
- Management



Code Roller™

- requirements management
- change management
- configuration management
- release management
- records management
- compliance management
- content management



Code Roller™

The screenshot displays the Code Roller web application interface. At the top, the logo and name 'Code Roller' are on the left, and navigation links 'Reviews Tasks Defects Profile Help' are on the right. A breadcrumb trail shows the current path: 'Home > Project > Package > Requirements > Use Case View > Test Step'. The main content area is titled 'Test Step' and is currently empty. The interface is divided into several sections: a left sidebar with 'Workspace' and 'Opportunity' categories, a right sidebar with 'Recently Visited', 'Projects', 'Teams', and 'Calendar' sections, and a footer with copyright information. Red numbers 1 through 10 are overlaid on the interface to highlight specific elements.

1 Reviews Tasks Defects Profile Help

2 Code Roller

3 Home > Project > Package > Requirements > Use Case View > Test Step

4 Workspace

- Requirements
- Use Cases
- Designs
- Test Plans
- Diagrams
- A Tester's Life
- Clad

5 Opportunity

- Blogs**
Created by a...
Blogging about
- People**
See other minded
people to help you
with...
- Projects**
See projects to get
with...
- Teams**
See our teams that you
can help...
- Process**
See how we do
successful software
development

6 Recently Visited

- Defect
- TestPlan
- TestStep

7 Projects

- bloggy
- winwin
- Start a Project

8 Teams

- wood
- earth in
- Start a Team

9 Calendar

- 2008-10-18
- Schedule an Event

10 Copyright © 2008 Dynamical Software, Inc. All Rights Reserved
www.code-roller.com | Contact Us | About Us

Thank You

<http://code-roller.com>

<http://www.dynamicalsoftware.com>

glenn@dynamicalsoftware.com