

Agile Engineering *for Architects*

Become a systems engineer, not just a programmer!

Ryan Shriver > Managing Consultant

rshriver@ddig.com

Download this presentation at theagileengineer.com

Today's Lessons

- **Define problems, not just solve them!**
- **Not just “what” but “how well”**
- **Over-focus on FEATURES, under-focus on QUALITIES (even Agile)**
- **Start early with Qualities and evolve the architecture**

Today's Scope: Right Problem

Specifying
Measurable
Qualities

Design Ideas

Software
Architecture

Design
Patterns

Tools

Coding
Practices

Refactoring

TDD

CI

**Problem
Right**

Agile
Integration

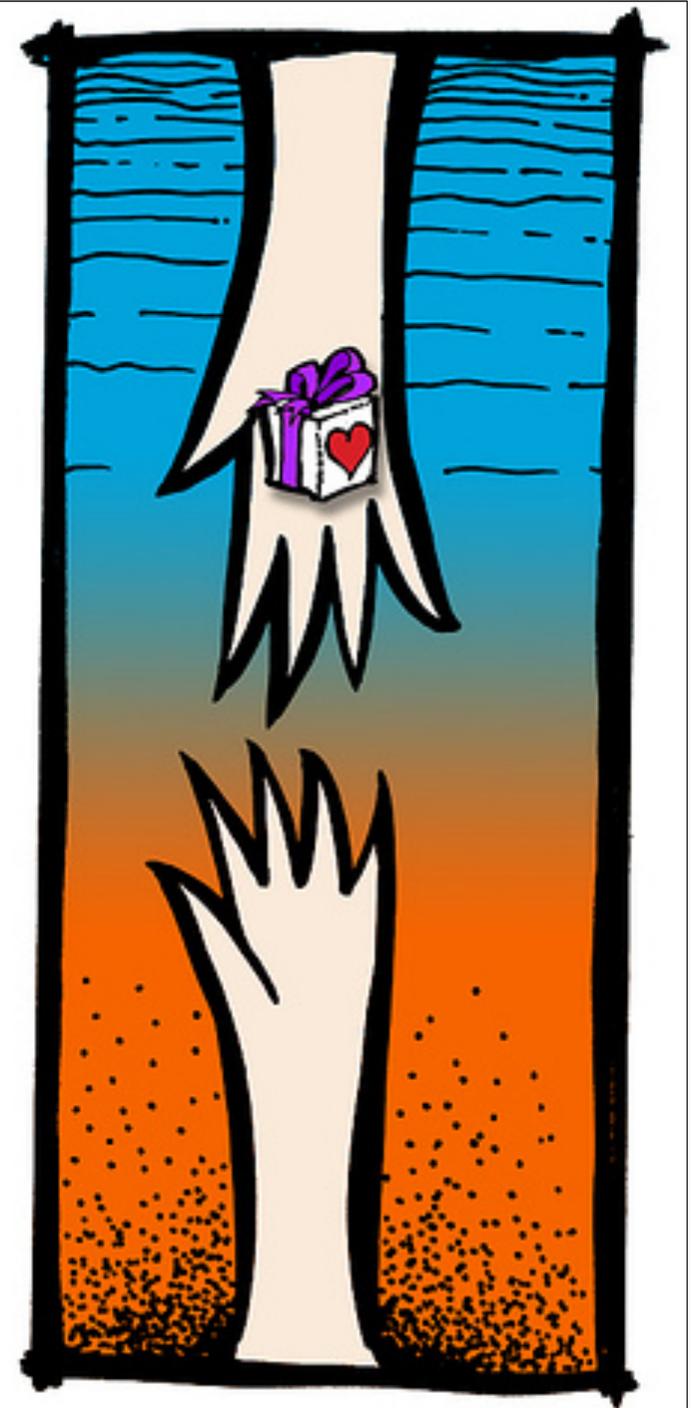
Next Best
Design Idea



Specifying Measurable Qualities

Our Client

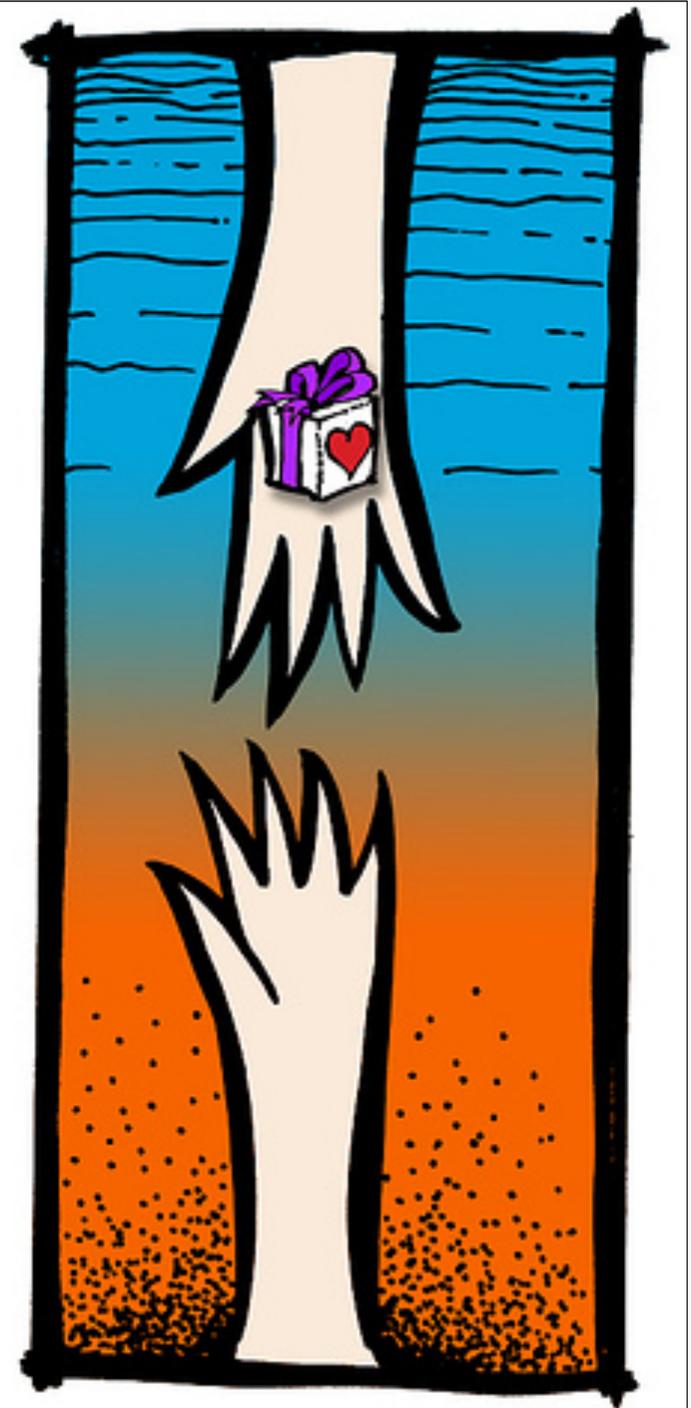
- Non Profit Organization
- Provides information on 1.7 million Non Profits in USA (money in/out, goals, board)
- Free and Paid Subscribers access information through web site
- Can research organizations, donate money and time



2009 Strategy

“The new 2.0 version of our products will provide our customers with an improved user experience and performance. This release is targeted for high-volume enterprise customers, a key segment for us to capture to continue our market growth.”

- CEO



Where to Start?

“The new 2.0 version of our products will provide our customers with an improved user experience and performance. This release is targeted for high-volume enterprise customers, a key segment for us to capture to continue our market growth.”

- CEO



Stakeholders

Possible System Qualities?

“The new 2.0 version of our products will provide our customers with an improved user experience and performance. This release is targeted for high-volume enterprise customers, a key segment for us to capture to continue our market growth.”

- CEO

- ★ Usability
- ★ Response Time
- ★ Availability
- ★ Scalability
- ★ Peak Throughput
- ★ Reliability
- ★ Recoverability

Exercise: Specify Qualities

Instructions: In small groups, using the system qualities below, spend 5 minutes identifying meaningful Scales and Meters.

- ★ Response Time
- ★ Availability
- ★ Peak Throughput

Scale: What to measure?

Meter: How to measure it?

Name: Recoverability

Example

Scale: Elapsed minutes between unexpected outage and fully available for use

Meter: Weekly monitoring report from independent data center polling application every 5 minutes

First Pass at Qualities

Name: Availability

Scale: Percentage of time system is available for accepting transactions excluding 1 hour weekly maintenance window

Meter: Weekly Monitoring Report from independent data center polling application every 5 minutes

Name: Response Time

Scale: Elapsed seconds between user requesting and receiving information for Specific Transactions

Meter: HP LoadRunner executing Standard Load scenario

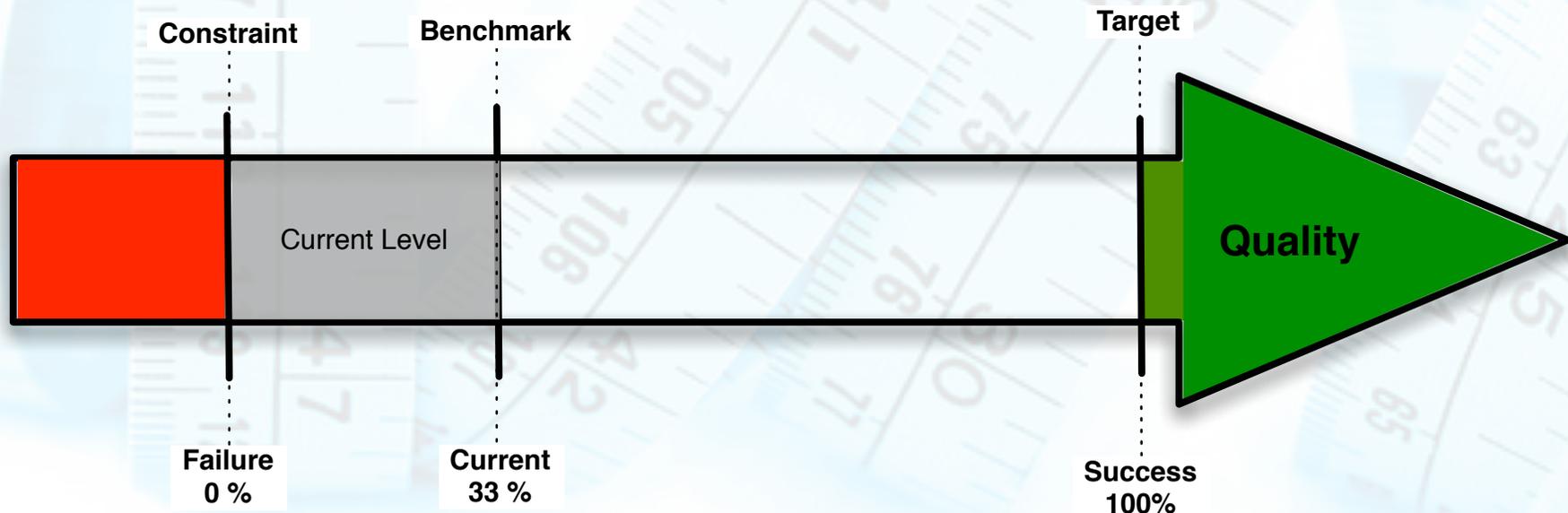
Name: Peak Throughput

Scale: Transactions per second with 90% of Response Times below acceptable levels for Specific Transactions

Meter: HP LoadRunner executing Standard Load scenario

Targets, Constraints & Benchmarks

Once we know “what” (scale) to measure and “how” (meter), our next step is to set achievement levels.



Final Pass at Qualities

Name: Availability

Scale: Percentage of time system is available for accepting transactions excluding 1 hour weekly maintenance window

Meter: Weekly Monitoring Report from independent data center polling application every 5 minutes

Target: $\geq 99.9\%$ (10 mins down)

Constraint: $< 99.4\%$ (60 mins down)

Benchmark [1.0]: 99.5% \leftarrow 2008 Q3 weekly average

Name: Peak Throughput

Scale: Transactions per second with 90% of Response Times below acceptable levels for Specific Transactions

Meter: HP LoadRunner executing Standard Load scenario

Target: ≥ 25

Constraint: < 20

Benchmark [1.0; Avg Top 5 Trans]: 15

Name: Response Time

Scale: Elapsed seconds between user requesting and receiving information for Specific Transactions

Meter: HP LoadRunner executing Standard Load scenario

Target: < 1 sec

Constraint: > 3 sec

Benchmark [1.0; Avg Top 5 Trans]: 2.8 secs

Summary

When specifying measurable qualities, focus on six attributes:

Name: Unique Quality Identifier

Scale: What to measure (units)

Meter: How to measure (method)

Target: Success level we're aiming to achieve

Constraint: Failure level we're seeking to avoid

Benchmark: Current level (or best in past)

Optionally:

- Use **[Qualifiers]** for specificity and re-usability
- Add <- Sources for transparency and credibility

What about User Stories?

Why not just document qualities as acceptance criteria on user stories?

- Sponsor & Architecture stakeholder visibility
- Separation of concerns (what vs. how well)
- Binary nature of acceptance criteria
- Stories should small enough to be completed in a Sprint

Today's Scope: Right Problem

Specifying
Measurable
Qualities



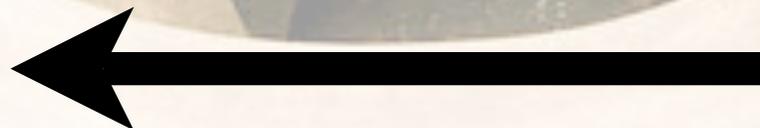
Design Ideas

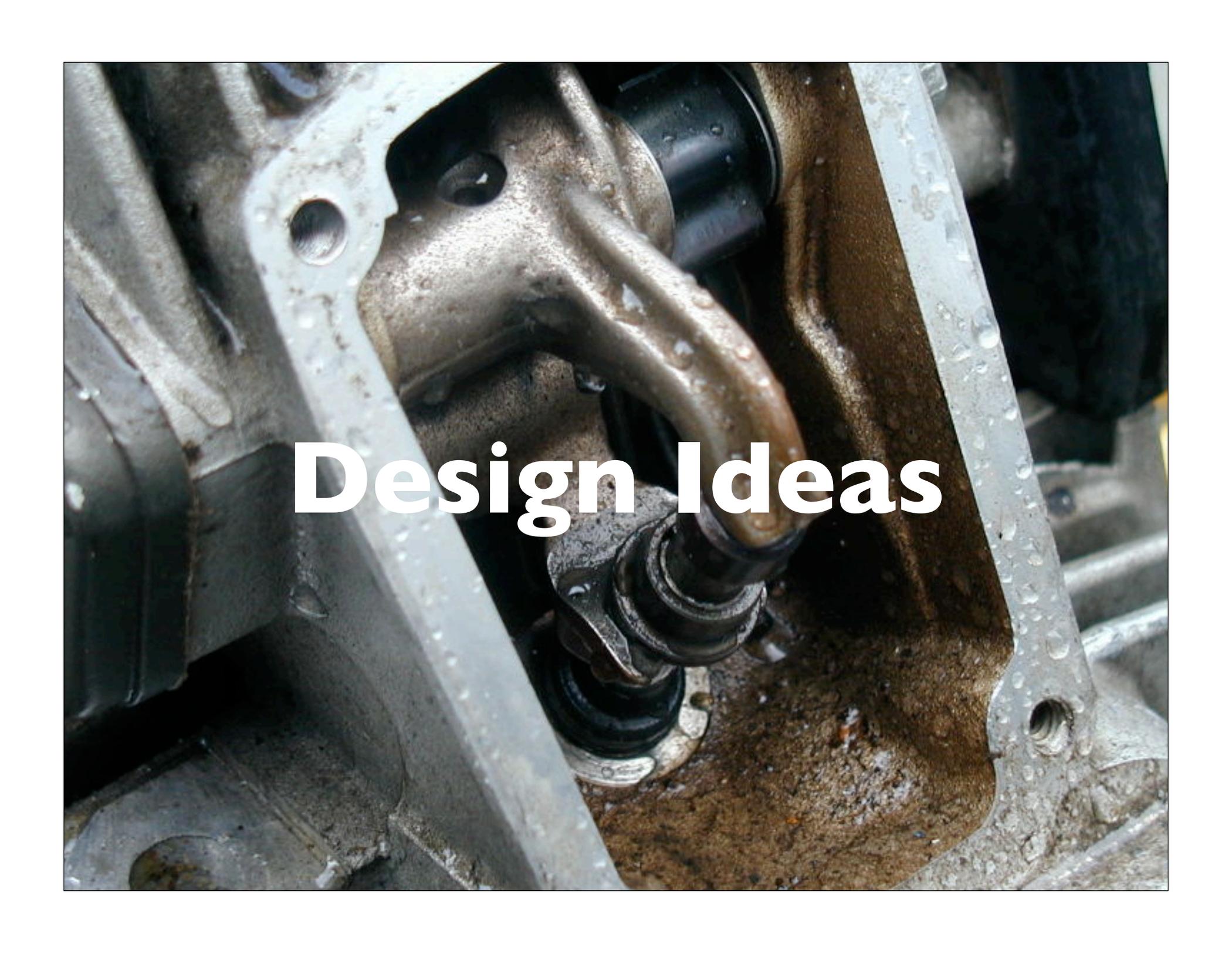


Agile
Integration



Next Best
Design Idea



A close-up photograph of a mechanical assembly, likely a bearing or a shaft component. The central shaft is dark and appears to be made of steel, with a bearing housing visible. The surrounding metal parts are light-colored, possibly aluminum, and show signs of wear and discoloration. The text "Design Ideas" is overlaid in a large, white, sans-serif font across the center of the image.

Design Ideas

From Qualities to Performance

Architects Should:

1. Specify Measurable Qualities
2. Help Stakeholders understand the Resources necessary to meet Target and Constraint Levels
3. Setup environments, processes, data and tools to assess Qualities

Architects should

Do By:

1. Benchmarking your system for important Qualities
2. Identifying the current bottleneck(s) limiting performance
3. Develop an plan to address the current bottleneck(s), with Impact Estimates

Be Agile!

Common Performance Bottlenecks

The bottleneck is somewhere



Exercise: Identify Design Ideas

Instructions: In same groups, spend 5 minutes identifying a few design ideas that could help improve Peak Throughput. Assume one of the four bottlenecks (CPU, Disk, Network, Memory)

Name: Peak Throughput

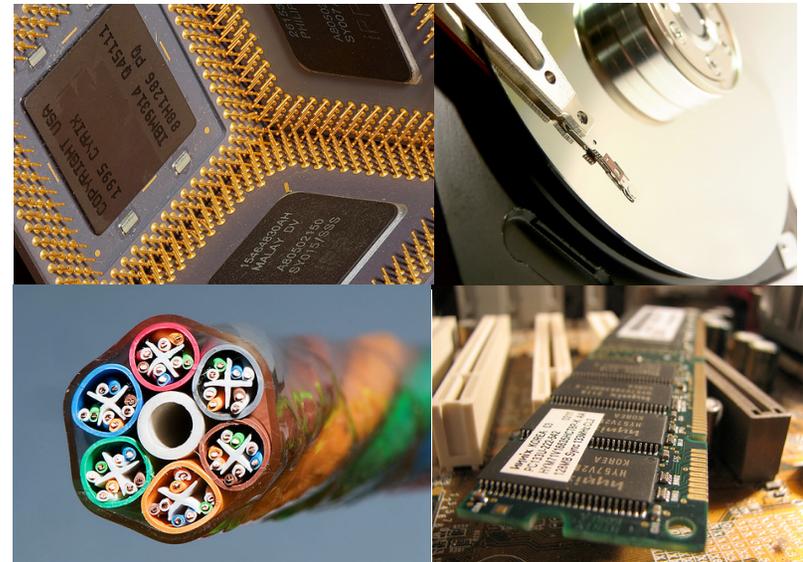
Scale: Transactions per second with 90% of Response Times below acceptable levels for Specific Transactions

Meter: HP LoadRunner executing Standard Load scenario

Target: ≥ 25

Constraint: < 20

Benchmark [1.0; Avg Top 5 Trans]: 15



Example Design Idea: CPU Bottleneck \Rightarrow Add Additional Servers

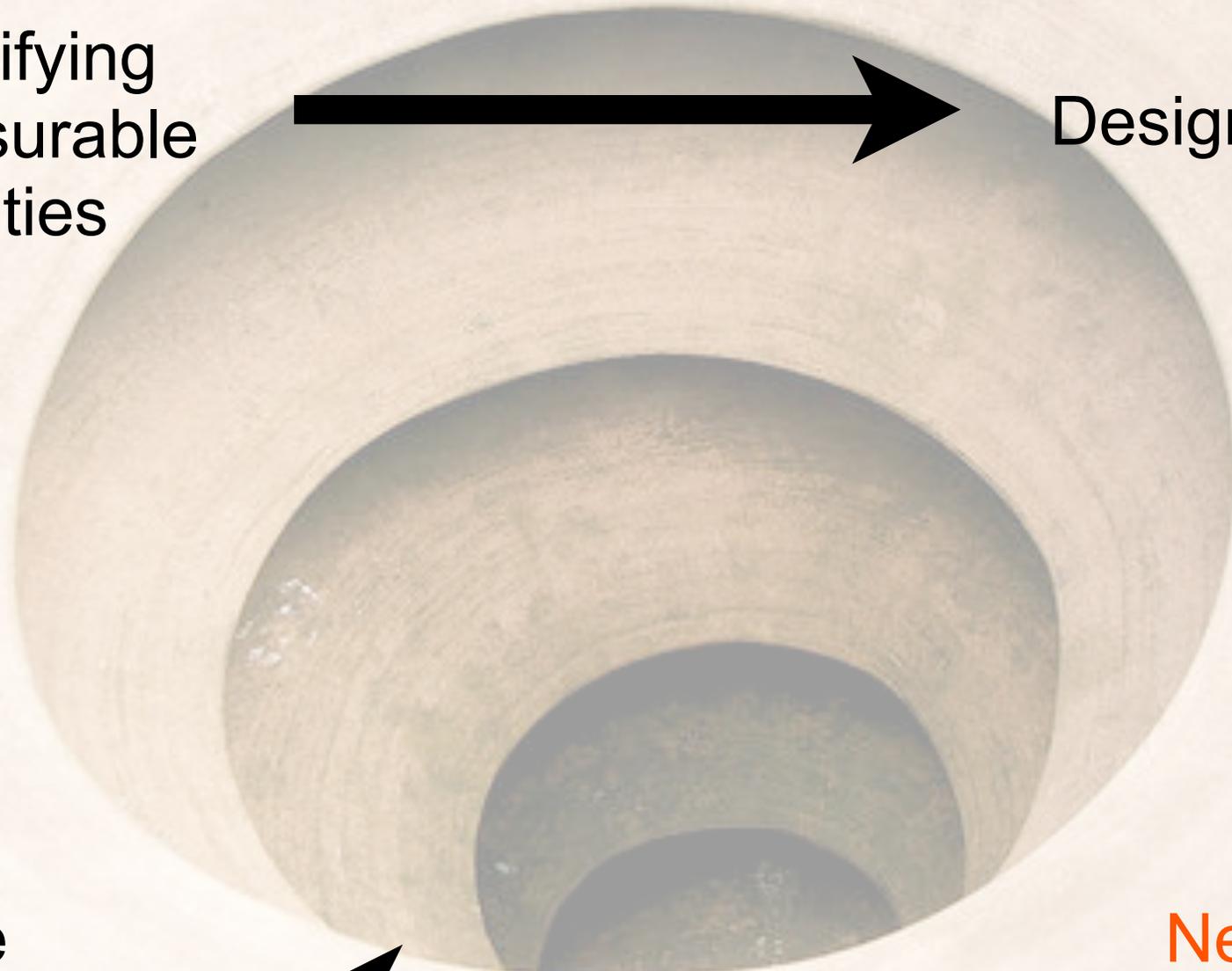
Today's Scope: Right Problem

Specifying
Measurable
Qualities

Design Ideas

Next Best
Design Idea

Agile
Integration





Next Best Design Idea

Next Best Design Idea

Impact Estimation Table	Design Idea	Design Idea
Qualities		
Resources		
Performance vs. Cost		



Impact Estimation Demo

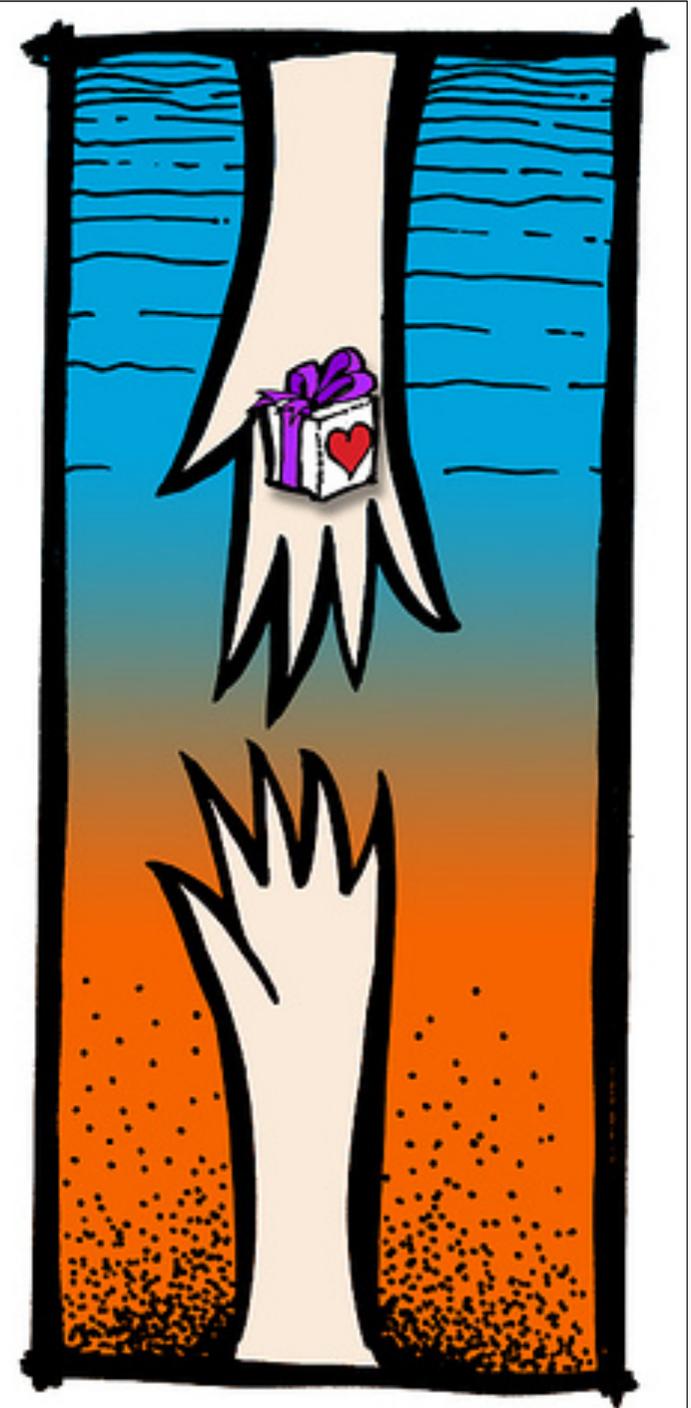
This Impact Estimation table can be downloaded from the Tools section of theagileengineer.com

	Recurring Payments	Facebook Integration	Image & Video Uploads	Total Impacts
Increase Market Share (6% to 10%)	30% +/- 20%	30% +/- 20%	20% +/- 10%	80% +/- 50%
Increase Monetary Donations (\$13M to \$18M)	80% +/- 30%	30% +/- 30%	50% +/- 20%	160% +/- 80%
Increase Volunteer Time Donations (2,700 to 3,600)	10% +/- 10%	50% +/- 20%	80% +/- 20%	140% +/- 70%
Total Objectives Impact	120% +/- 60%	110% +/- 70%	150% +/- 50%	
Money (\$1.0M)	30% +/- 10%	20% +/- 10%	50% +/- 20%	100% +/- 40%
Time (10 months)	40% +/- 20%	20% +/- 10%	50% +/- 10%	110% +/- 60%
Total Budget Impact	70% +/- 30%	40% +/- 20%	100% +/- 30%	
Benefit / Cost Ratio	120/70 = 1.7 Worse: 0.6 Best: 4.5	110/40 = 2.8 Worse: 0.7 Best: 9	150/100 = 1.5 Worse: 0.9 Best: 2.9	

Next Best Design?

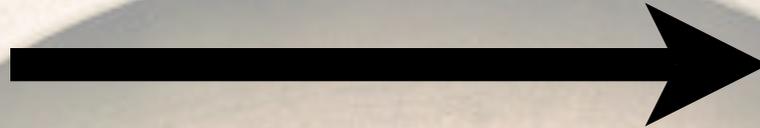
Which Design Idea should our client pursue next?

What would be some of the tasks associated with implementing the design?



Today's Scope: Right Problem

Specifying
Measurable
Qualities



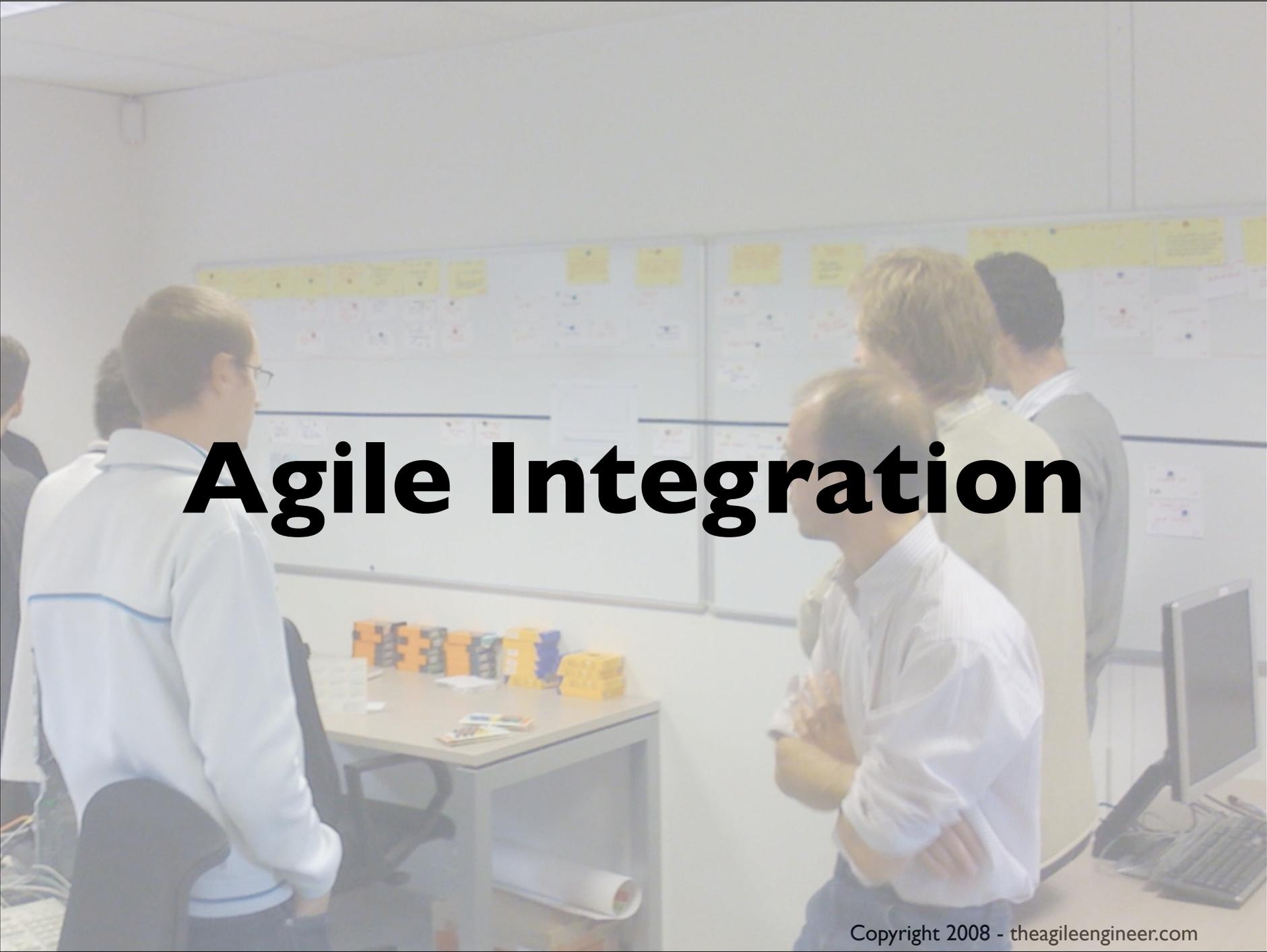
Design Ideas



Next Best
Design Idea

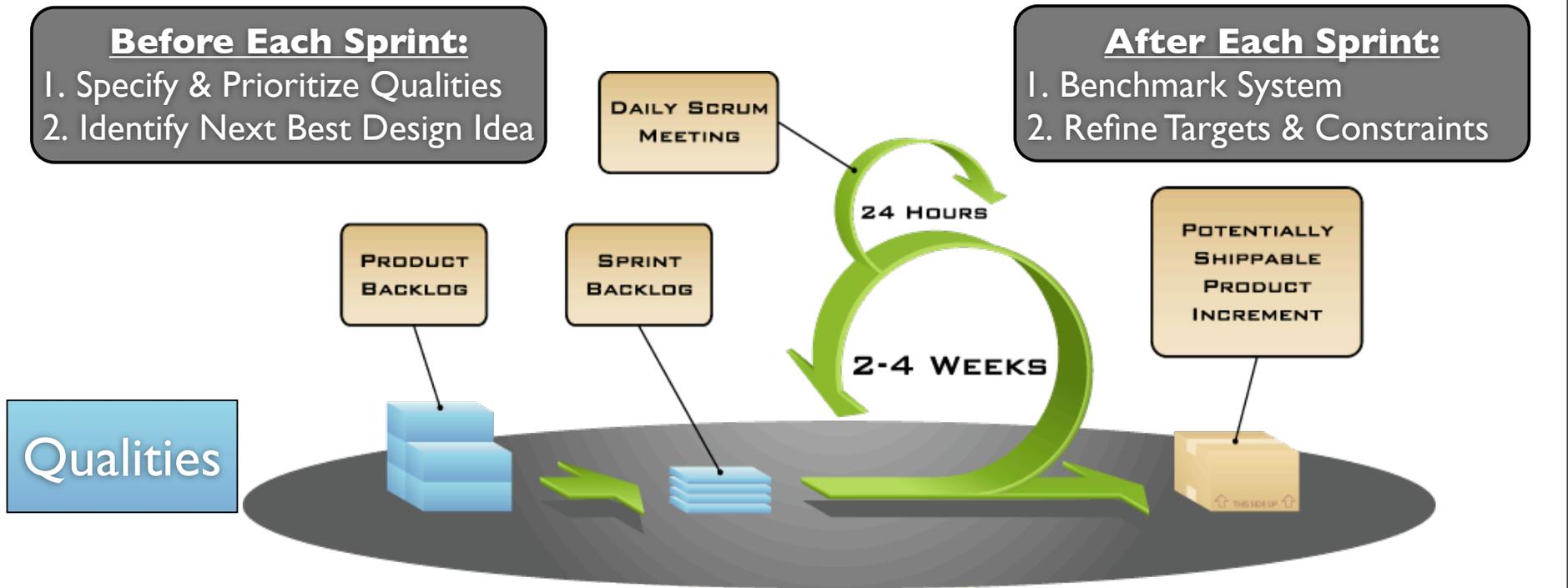


Agile
Integration

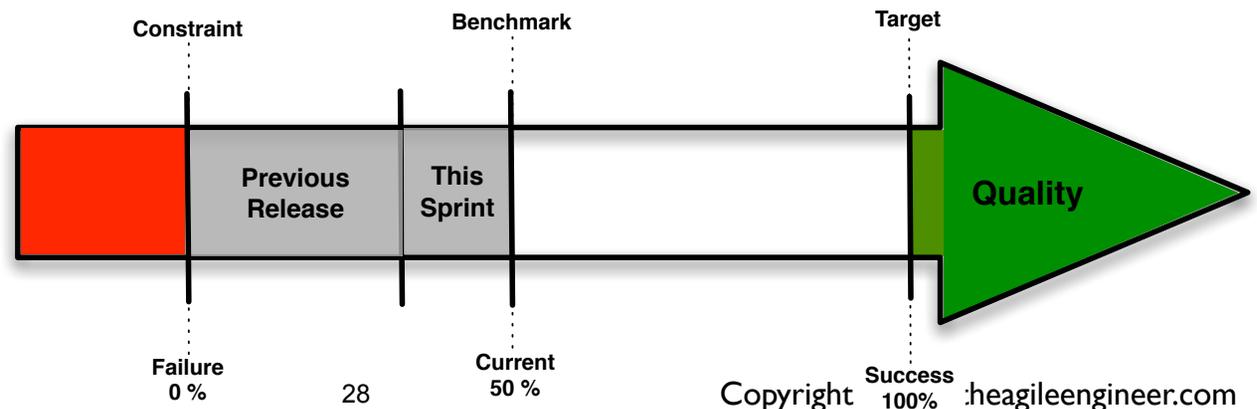
A group of people in a meeting room looking at a whiteboard with sticky notes. The room is brightly lit, and the whiteboard is covered with various documents and sticky notes. The people are dressed in business casual attire. The overall atmosphere is professional and collaborative.

Agile Integration

Agile Integration



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE



Summary

- **Define problems using Measurable Qualities**
- **Negotiate “how well” with Stakeholders**
- **Use Impact Estimation to find “best bang for the buck”**
- **Start early with Measurable Qualities and evolve the Architecture**

Next Steps

1. Try it out and get **FREE** support!
2. Sign up for Open Source tool in 2009
3. Stay in touch at theagileengineer.com
 - i. *“Are we feature builders or value delivers”*
 - ii. *“Qualities, User Stories and the sad state of Scrum Requirements”*

Thank You!

Become a systems engineer, not just a programmer!

Ryan Shriver > Managing Consultant

rshriver@ddig.com

Download this presentation at theagileengineer.com