

springone **2GX**

**Chicago, October 19 - 22,
2010**

Slimmed Down Software A Lean, Groovy Approach

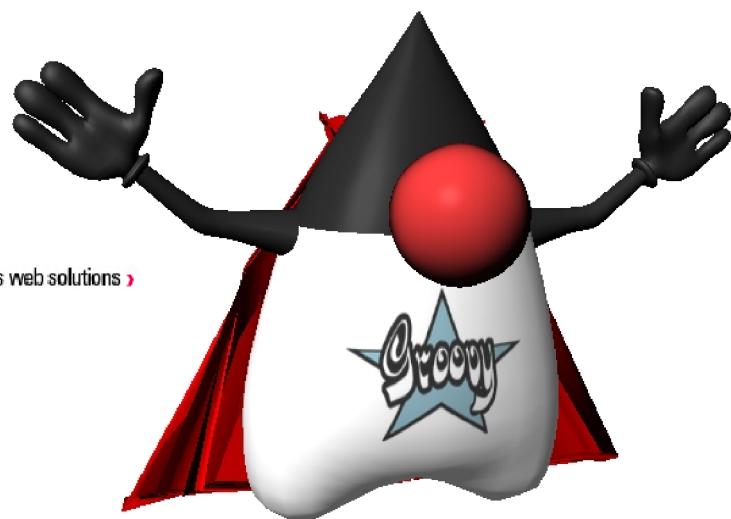
Hamlet D'Arcy
Canoo Engineering AG
@HamletDRC

SpringOne 2GX 2010. All rights reserved. Do not distribute without permission.



canoo

› your provider for business web solutions ›



GROOVY MAG

for groovy and grails developers



<http://groovymag.com>

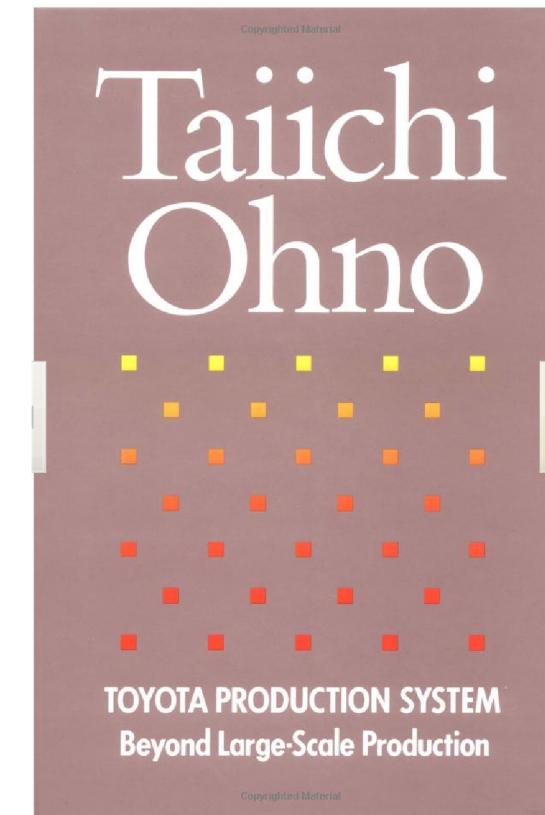
A screenshot of a web browser displaying a blog post from the 'canoo' blog. The title of the post is 'Slimmed Down Software – A Lean, Groovy Approach Part 1 – Eliminate Waste'. The post content discusses the Groovy Programming Language and its application in software development. Below the post is a section titled 'About This Series' with a brief description of the series.

<http://canoo.com/blog>

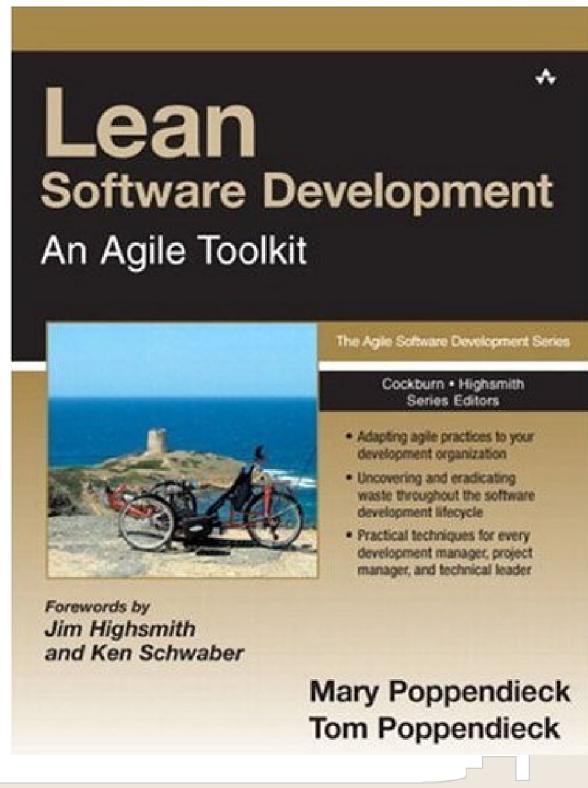


<http://nofluffjuststuff.com/>

1988 - Japan



2003 - Minnesota



Lean Principles

- Eliminate Waste
- Build Quality In
- Create Knowledge
- Defer Commitment
- Deliver Fast
- Respect People
- Optimize the Whole

Lean Principles

- Eliminate Waste
- Build Quality In
- Create Knowledge
- Defer Commitment
- Deliver Fast
- Respect People
- Optimize the Whole

XP Values

- Communication
- Simplicity
- Feedback
- Courage
- Respect

Lean Principles

- Eliminate Waste
- Build Quality In
- Create Knowledge
- Defer Commitment
- Deliver Fast
- Respect People
- Optimize the Whole

XP Values

- Communication
- Simplicity
- Feedback
- Courage
- Respect

XP Principles

- Feedback
- Assuming Simplicity
- Embracing Change

Lean Principles

- Eliminate Waste
- Build Quality In
- Create Knowledge
- Defer Commitment
- Deliver Fast
- Respect People
- Optimize the Whole

Lean Principles

- **Eliminate Waste**
- Build Quality In
- Create Knowledge
- Defer Commitment
- Deliver Fast
- Respect People
- Optimize the Whole

Eliminate Waste?

```
Map m = [1: 'a', 2: 'b', 3 : 'c']
```

versus

```
Map m = new HashMap();
m.put(1, "a");
m.put(2, "b");
m.put(3, "c");
```

Eliminate Waste?

```
Map m = [1: 'a', 2: 'b', 3 : 'c']
```

```
Map m = new HashMap<Integer, String>();
```

```
m.put(1, "a");  
m.put(2, "b");  
m.put(3, "c");
```

Phase 1

stop doing
things that
aren't worth
the effort

Phase 2

do the same
things with
less effort

Phase 3

Profit!



12

springone 2GX

Membra Them?!



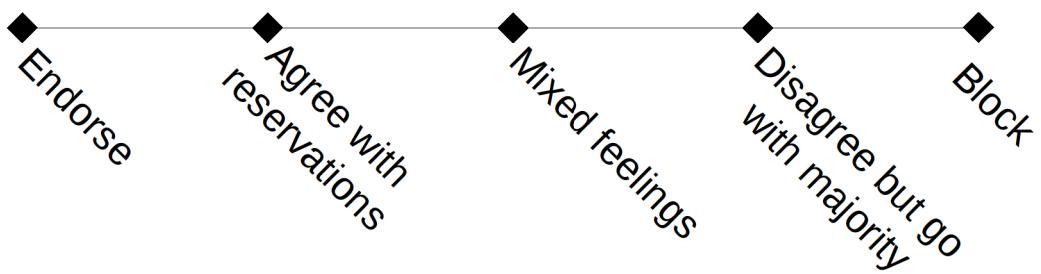
springone **20X**

Deja Vu?



How to Make a Decision

- Consensus
- Majority
- Plurality
- Dictatorship
- *Gradients of Agreement*



From “Agile Coaching”

Write Only Process Improvement

Working software over
comprehensive documentation
as long as that software is
comprehensively documented.

... from <http://www.halfarsedagilemanifesto.org>

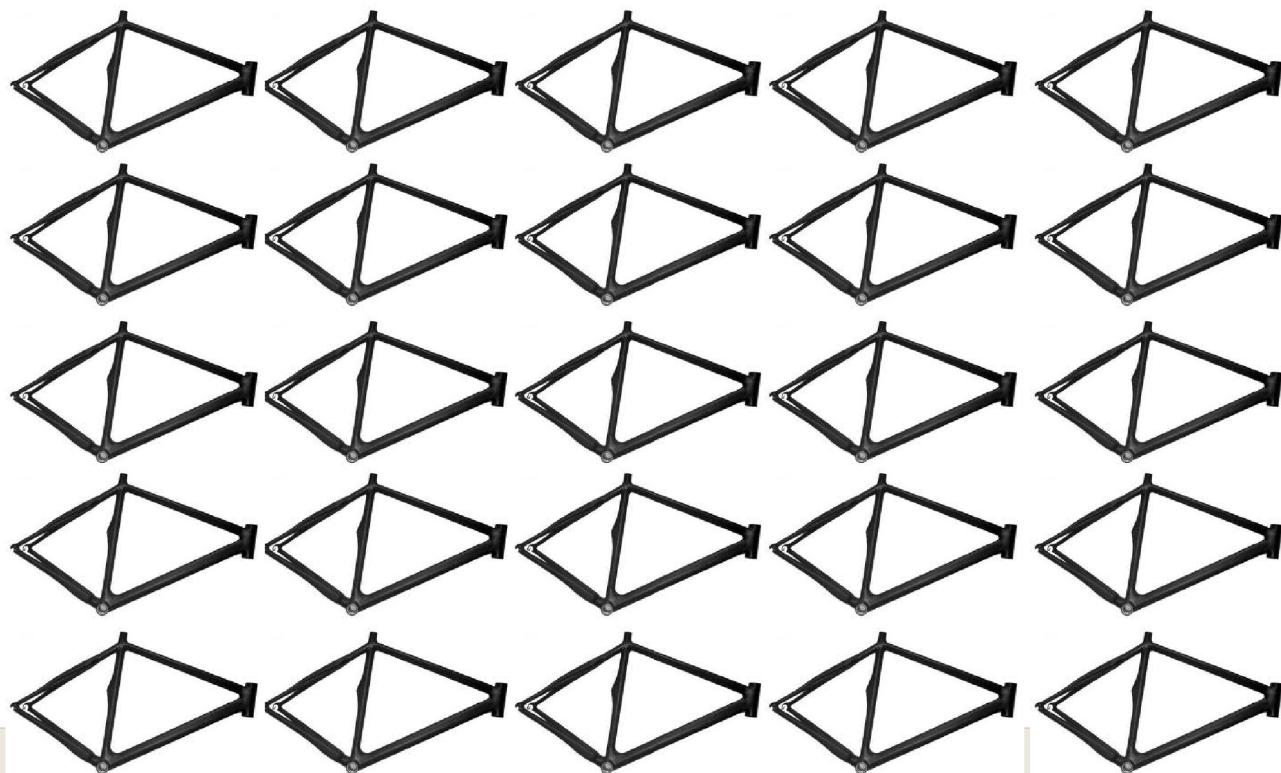
Recognizing Waste

- In-Process Inventory
- Over-Production
- Extra Processing
- Transportation
- Motion
- Waiting
- Defects

Recognizing Waste

- Partially Done Work (In-Process Inventory)
- Extra Features (Over-Production)
- Relearning (Extra Processing)
- Handoffs (Transportation)
- Task Switching (Motion)
- Delays (Waiting)
- Defects (Defects)

Partially Done Work



Partially Done Work

```

<ns1:run xmlns="http://www.w3.org/2005/05/xpath-operators">
  <ns1:step name="TextSearchBinding" binding="TextSearchBinding">
    <ns1:method href="http://schemas.microsoft.com/nisearch/api/search/http"/>
    <ns1:operation name="Add">
      <ns1:source type="Document" nsid="http://episerver.com/valibyly"/>
      <ns1:target type="Document" nsid="http://episerver.com/valibyly"/>
      <ns1:properties encoding="http://schemas.microsoft.com/soap/encoding/">
        <ns1:property name="Add">
          <ns1:source href="http://search/internet.org" type="Text"/>
          <ns1:target href="http://search/internet.org" type="Text"/>
        </ns1:property>
      </ns1:properties>
    </ns1:operation>
  </ns1:step>
  <ns1:step name="Add" soapAction="http://sapinrep.org/Add">
    <ns1:operation name="Add">
      <ns1:source type="Document" nsid="http://episerver.com/valibyly"/>
      <ns1:target type="Document" nsid="http://episerver.com/valibyly"/>
      <ns1:properties encoding="http://schemas.microsoft.com/soap/encoding/">
        <ns1:property name="Add">
          <ns1:source href="http://search/internet.org" type="Text"/>
          <ns1:target href="http://search/internet.org" type="Text"/>
        </ns1:property>
      </ns1:properties>
    </ns1:operation>
  </ns1:step>
  <ns1:step name="Delete" soapAction="http://sapinrep.org/Delete">
    <ns1:operation name="Delete" type="TextSearchBinding">
      <ns1:method href="http://schemas.microsoft.com/nisearch/api/deleteDocAndRsp"/>
      <ns1:args>
        <ns1:arg name="DeleteDocAndRsp">
          <ns1:source type="TextSearchBinding" binding="TextSearchBinding"/>
        </ns1:arg>
      </ns1:args>
    </ns1:operation>
  </ns1:step>
</ns1:run>

```

```

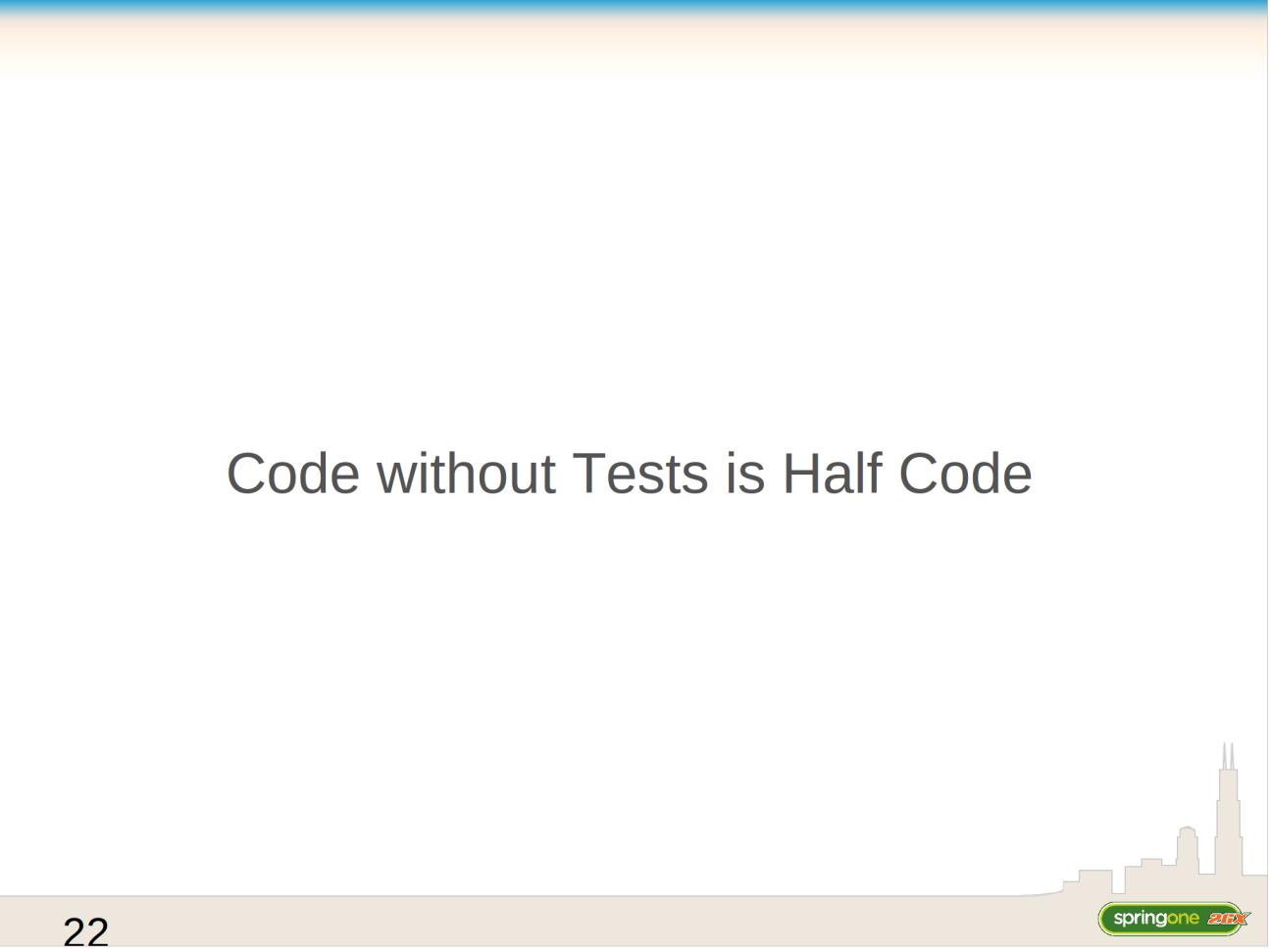
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <wsse:Security soap:mustUnderstand="1" xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/12/ws-security"/>
      <wsse:UsernameToken wsse:mustUnderstand="1">
        <wsse:Username>TestUser</wsse:Username>
        <wsse:Password>TestUser123</wsse:Password>
        <wsse:Nonce>DwJLjWzqQHdXGZBfPnVYQ==</wsse:Nonce>
        <wsse:Created>2012-07-10T11:45:00Z</wsse:Created>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <ns1:UpdateAndSyncRequest
      xmlns:ns1="http://schemas.xmlsoap.org/soap/encoding/">
      <ns1:operation name="UpdateSync" type="xsd:string">
        <ns1:parameter name="syncId" type="xsd:string">10000000000000000000000000000000</ns1:parameter>
        <ns1:parameter name="syncData" type="xsd:string"><ns1:content>{"syncId": "10000000000000000000000000000000", "syncData": "10000000000000000000000000000000", "syncType": "Sync", "syncStatus": "Success", "syncTime": "2012-07-10T11:45:00Z"}</ns1:content></ns1:parameter>
      </ns1:operation>
    </ns1:UpdateAndSyncRequest>
  </soap:Body>
</soap:Envelope>

```

```

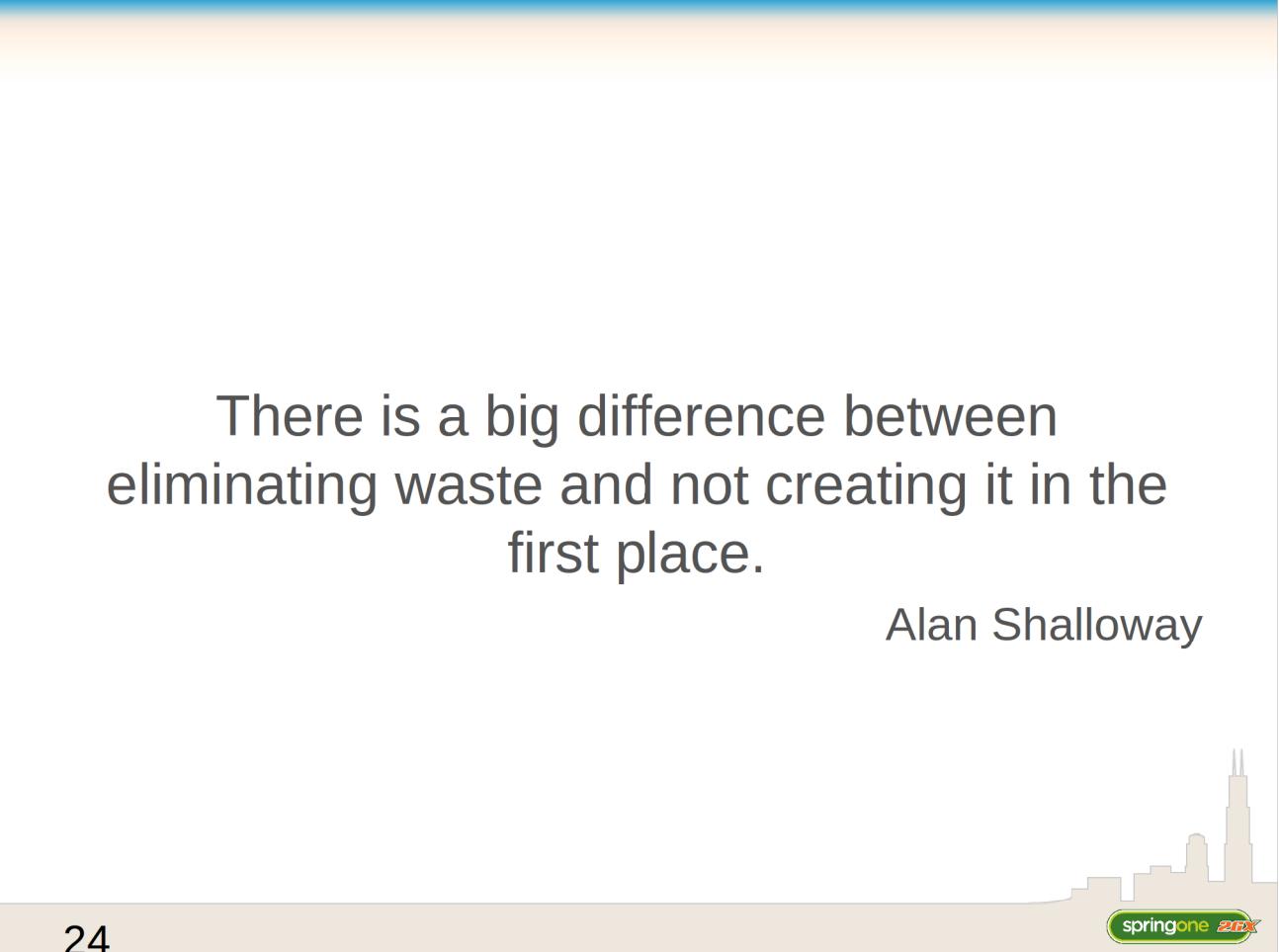
<ns1:ns1 name="TestSequence1" type="ns1:ns1SequenceType">
  <ns1:ns1Item name="TestSequence1Item1">
    <ns1:ns1Content href="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="HelloWorld"/>
    <operation name="Add"/>
    <operation name="Subtract"/>
    <operation name="Multiply"/>
    <operation name="Divide"/>
  </ns1:ns1Item>
  <ns1:ns1Body encoding="http://schemas.xmlsoap.org/soap/encoding" />
</ns1:ns1>
<ns1:ns1Item name="TestSequence1Item2">
  <ns1:ns1Content href="http://schemas.xmlsoap.org/soap/encoding" />
  <ns1:ns1Operation name="Add" type="ns1:ns1OperationType">
    <ns1:ns1Document soapAction="http://appleipr.org/Add" />
    <ns1:ns1Input />
    <ns1:ns1Body use="Literal" />
    <ns1:ns1Output />
    <ns1:ns1Fault />
  </ns1:ns1Operation>
  <ns1:ns1Operation name="Subtract" type="ns1:ns1OperationType">
    <ns1:ns1Document soapAction="http://appleipr.org/Subtract" />
    <ns1:ns1Input />
    <ns1:ns1Body use="Literal" />
    <ns1:ns1Output />
    <ns1:ns1Fault />
  </ns1:ns1Operation>
  <ns1:ns1Operation name="HelloWorld" type="ns1:ns1OperationType">
    <ns1:ns1Document soapAction="http://appleipr.org/HelloWorld" />
    <ns1:ns1Input />
    <ns1:ns1Body use="Literal" />
    <ns1:ns1Output />
    <ns1:ns1Fault />
  </ns1:ns1Operation>
</ns1:ns1Item>
</ns1:ns1>
<ns1:ns1Item name="HelloWorld" type="ns1:ns1Service">
  <ns1:ns1Name value="HelloWorld" />
  <ns1:ns1Address href="http://localhost/services/HelloWorldDockerRipC" />
</ns1:ns1Item>
</ns1:ns1>

```



Code without Tests is Half Code

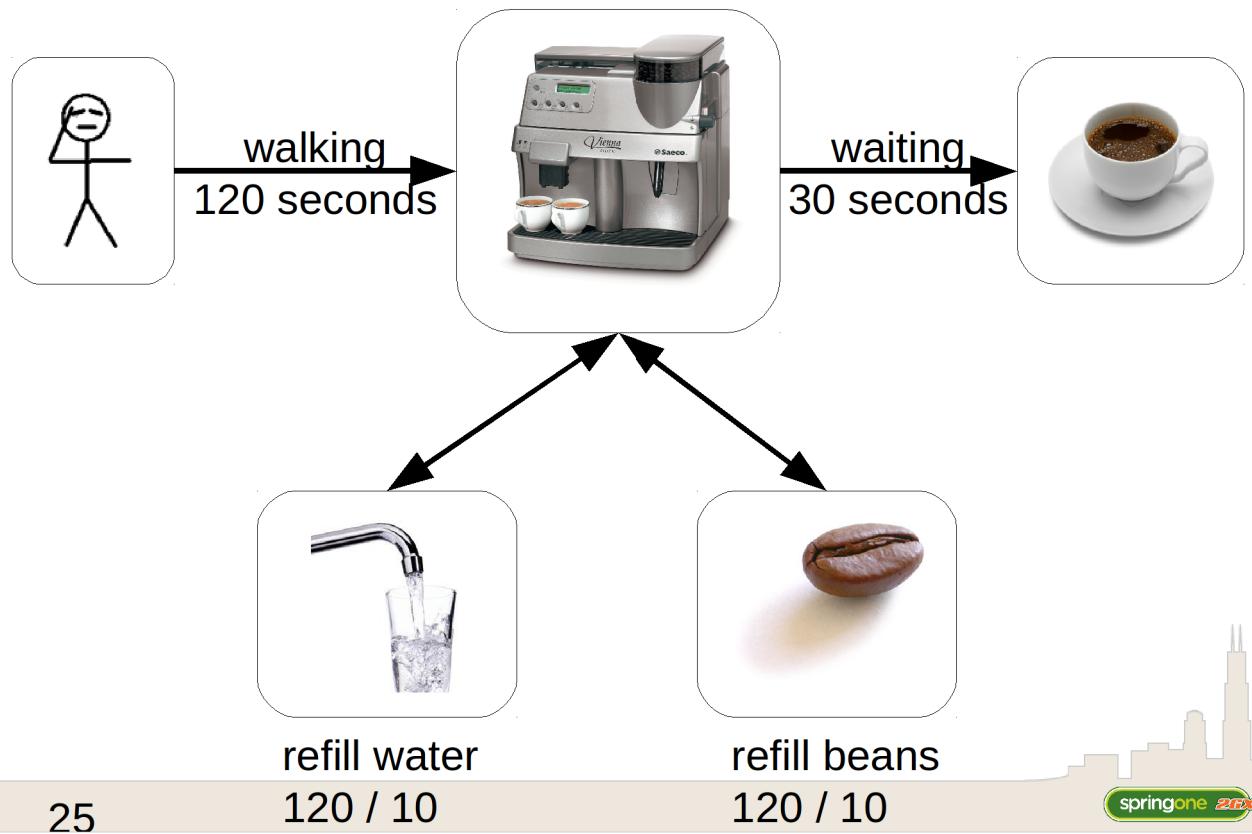
Demo



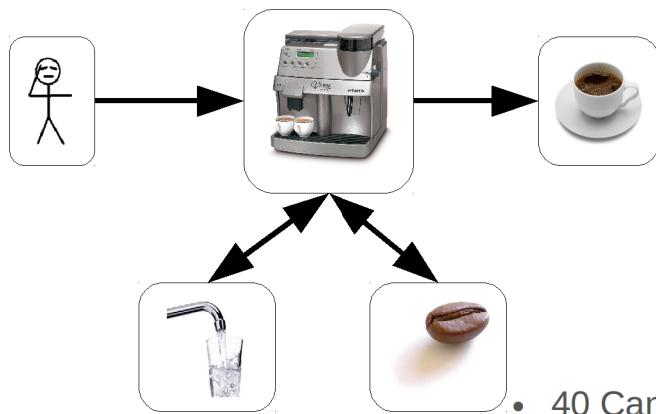
There is a big difference between
eliminating waste and not creating it in the
first place.

Alan Shalloway

Next Steps: Value Stream Mapping



Next Steps: Value Stream Mapping



- 40 Canoies
- 5 days a week
- 45 Weeks a Year
- 2 Cups a Day
- *174 Seconds Average*
- \$50 an Hour
= **\$43,500 a year**

- 40 Canoies
- 5 days a week
- 45 Weeks a Year
- 2 Cups a Day
- *150 Seconds Average*
- \$50 an Hour
= **\$37,500 a year**

Need more waste?



The Seven Wastes of Software Development - Agile Zone - Google Chrome

dz The Seven Wastes of Software Development - Agile Zone - Google Chrome

zimbra Bookmarks Kickass Other Bookmarks

AGILE ZONE

Home Zones Library Refcardz Forums Links ITQuestions Snippets

About the author

Eric Helmick Last Seen 5m ZDNet Author Profile Edit Journal Logins View all 3 posts

Recommended Links

- Saving Agile Within Your Organization
- The Agile Project Manager
- The Agile Coach
- Agile Project Management Tools
- Agile Business Methods

Like this post? Leave a comment below or share this on social media.

VersionOne

Read the other parts in the series:

- The Seven Wastes of Software Development: Introduction
- Waste #1 - Waiting for Work
- Waste #2 - Extra Features
- Waste #3 - Relearning
- Waste #4 - Defects
- Waste #5 - Task Switching
- Waste #6 - Delays

To understand Dev's focus on waste in iteration, you must first understand a bit of the process focus for Devs in the late 1990s. Manufacturing has always been an expensive process, and it's one that can be held hostage. And yet the typical outcome for Devs in the late 1990s was a great deal of waste. For Devs to succeed, they had to reduce the types of waste by reducing the cost of their software.

Eric's TPS was the other way. This is how Eric described the essence of the TPS:

All we're doing is getting off the Internet from the moment a customer gives us an order to the point when we collect the cash. And we're reducing that timeline by removing the non-value-added wastes [1].

Popular at DZone

HyperText Markup Language (HTML) - Help Center - Help Center

Get the latest from the Web

Graviton in Business Software: Why It's Red

Don't Implement Web Development by Yourself - David

Don't let UI design affect UI tests

New Zend Grid Charts Encourage Bad JavaScript

Help in Part 1

3 Simple, Critical Steps for Project Management

See more popular at DZone

Subscribe to the RSS feed

springone 2007

Lean Principles

- Eliminate Waste
- Build Quality In
- Create Knowledge
- Defer Commitment
- Deliver Fast
- Respect People
- Optimize the Whole

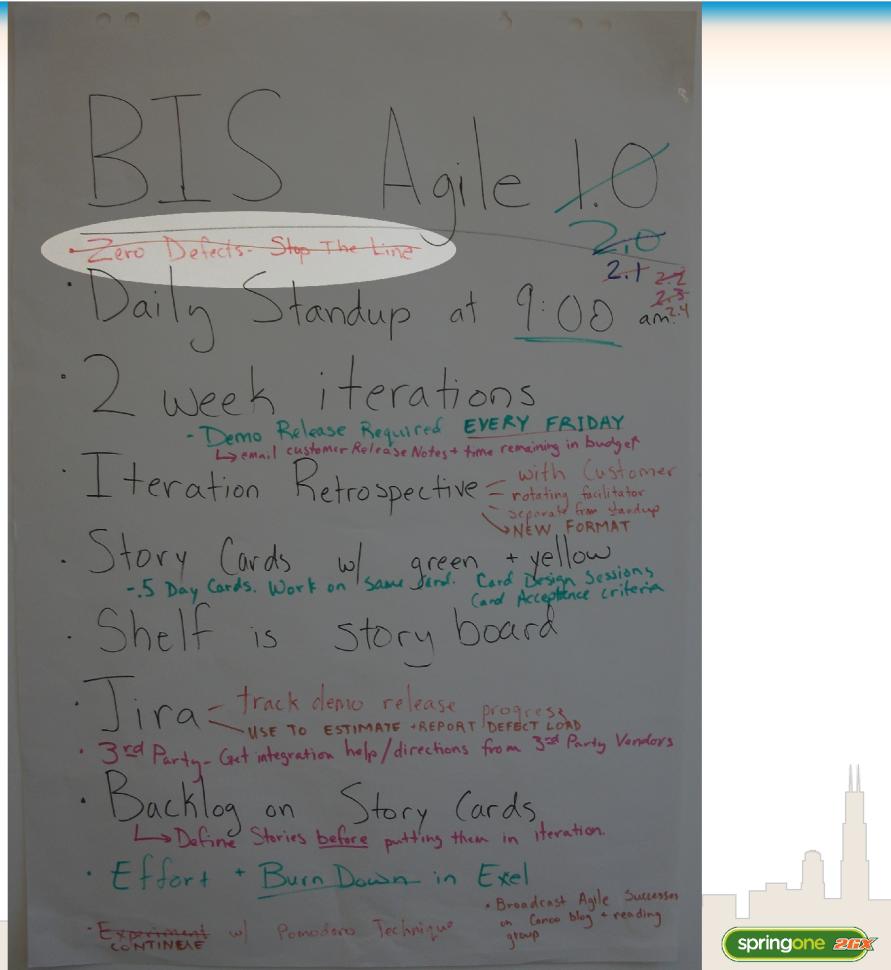


Demo



Really?!?: Behavior Driven Development

Next Steps: Zero Defects Stop the Line



Lean Principles

- Eliminate Waste
- Build Quality In
- **Create Knowledge**
- Defer Commitment
- Deliver Fast
- Respect People
- Optimize the Whole

Software Engineer
or
Computer Programmer?

Software Development as Knowledge Acquisition

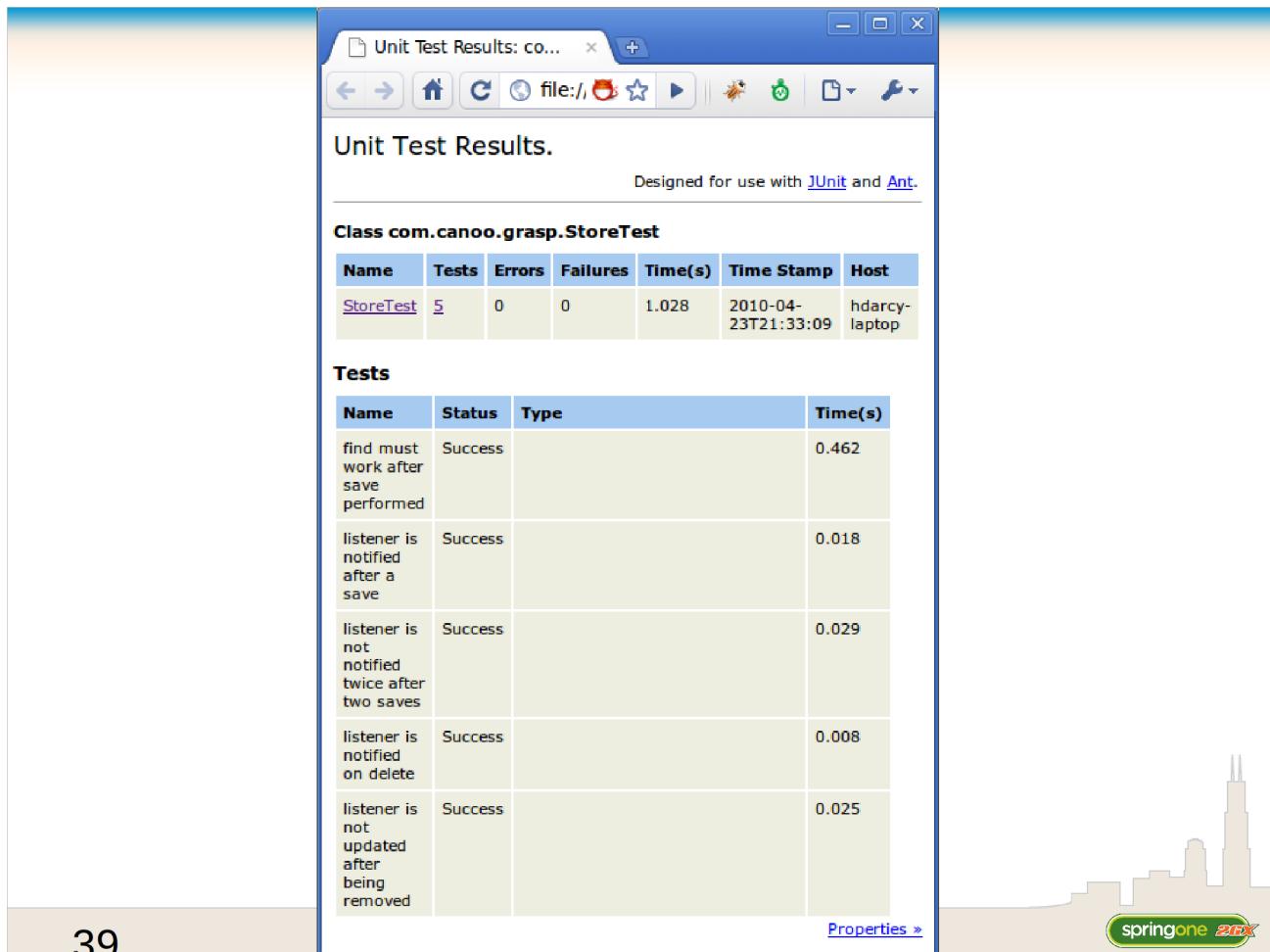
- short, frequent learning cycles
- delayed commitment

Large requirements and specification documents provide a wonderful illusion of certainty

Can tests be Documentation?

Evangelists have oversold
Developers have under-delivered

Demo



Lean Principles

- Eliminate Waste
- Build Quality In
- Create Knowledge
- **Defer Commitment**
- Deliver Fast
- Respect People
- Optimize the Whole



3 Types of Decisions

- a right decision
- a wrong decision
- no decision



springone 2GX

3 Types of Decisions

- a premature decision *today*
- an informed decision *tomorrow*
- no decision and *missed opportunity*

How will you get home tonight?

Last bus

Last train

Taxi

Stay at hotel

How will you get home tonight?

Last bus	\$4.00
Last train	\$8.00
Taxi	\$25.00
Stay at hotel	\$99.00

Options have value

How will you get home tonight?

Last bus	\$4.00	11 PM
Last train	\$8.00	12 PM
Taxi	\$25.00	All Night
Stay at hotel	\$99.00	All Night

*Options have value
Options expire*



springone ZRX

How will you get home tonight?

Last bus	\$4.00	11 PM
Last train	\$8.00	12 PM
Taxi	\$25.00	All Night
Stay at hotel	\$99.00	All Night

Options have value

Options expire

Never commit early unless you know why

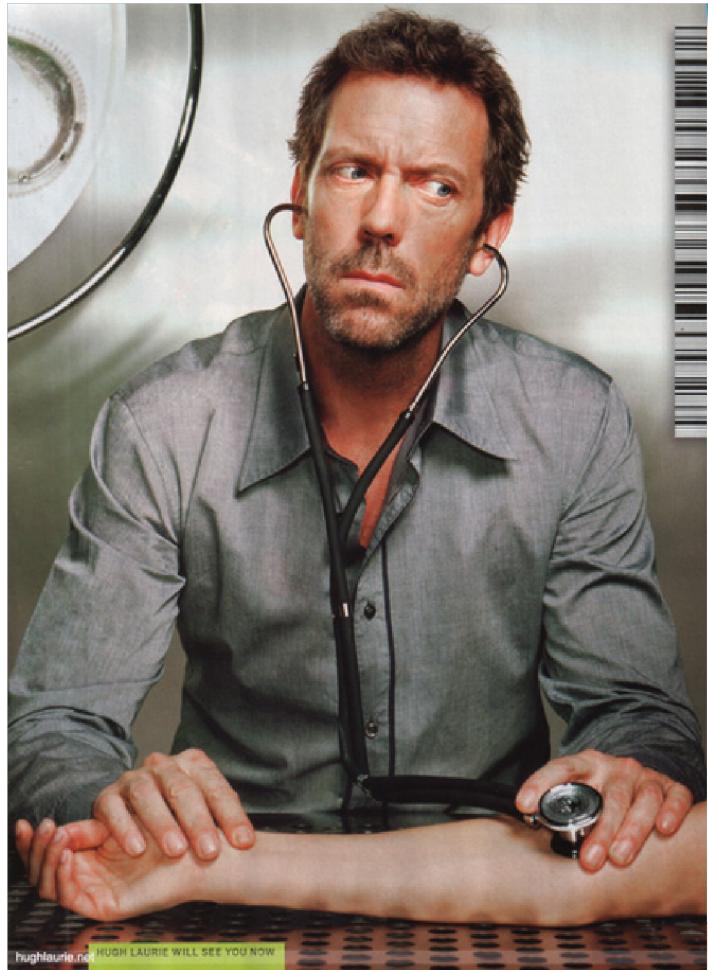


springone ZRX



LOLcaption.com

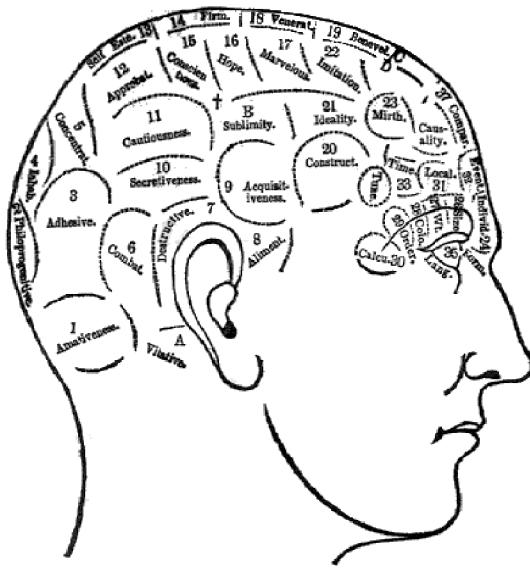
“The last responsible moment is after which execution of an option is too late. The option no longer exists or you’re not able to execute it any more.”



... weekly lesson in
delayed commitment



springone **200X**



Humans hate uncertainty.

So give them specifics.



“Very dynamic languages like Lisp, TCL and Smalltalk are often used for prototyping... [One] reason given for these languages being good for prototyping is that they don't require you to pin down decisions early on. JAVA has exactly the opposite property: it forces you to make choices explicitly.”

James Gosling 1996

Late Bound and Low Coupled

```
public void testUserView() {  
    def mockService = [  
        getUsers : { ... }  
    ]  
  
    def view = new UserView()  
    view.selectUser(1)  
    view.removeCurrent()  
    ...  
}
```



Is the value of an IT system in
the inputs or the outputs?

Groovy for Unit Testing

Deferring Commitment

An Undo-able Decision

Next Steps: Delayed Commitment Applied

- 1) Identify the options
- 2) Identify when the option expires
- 3) Identify steps to seek new options until the expiry date
- 4) Wait... and wait... and wait... until the conditions are correct for a decision
- 5) Decide quickly and act with confidence



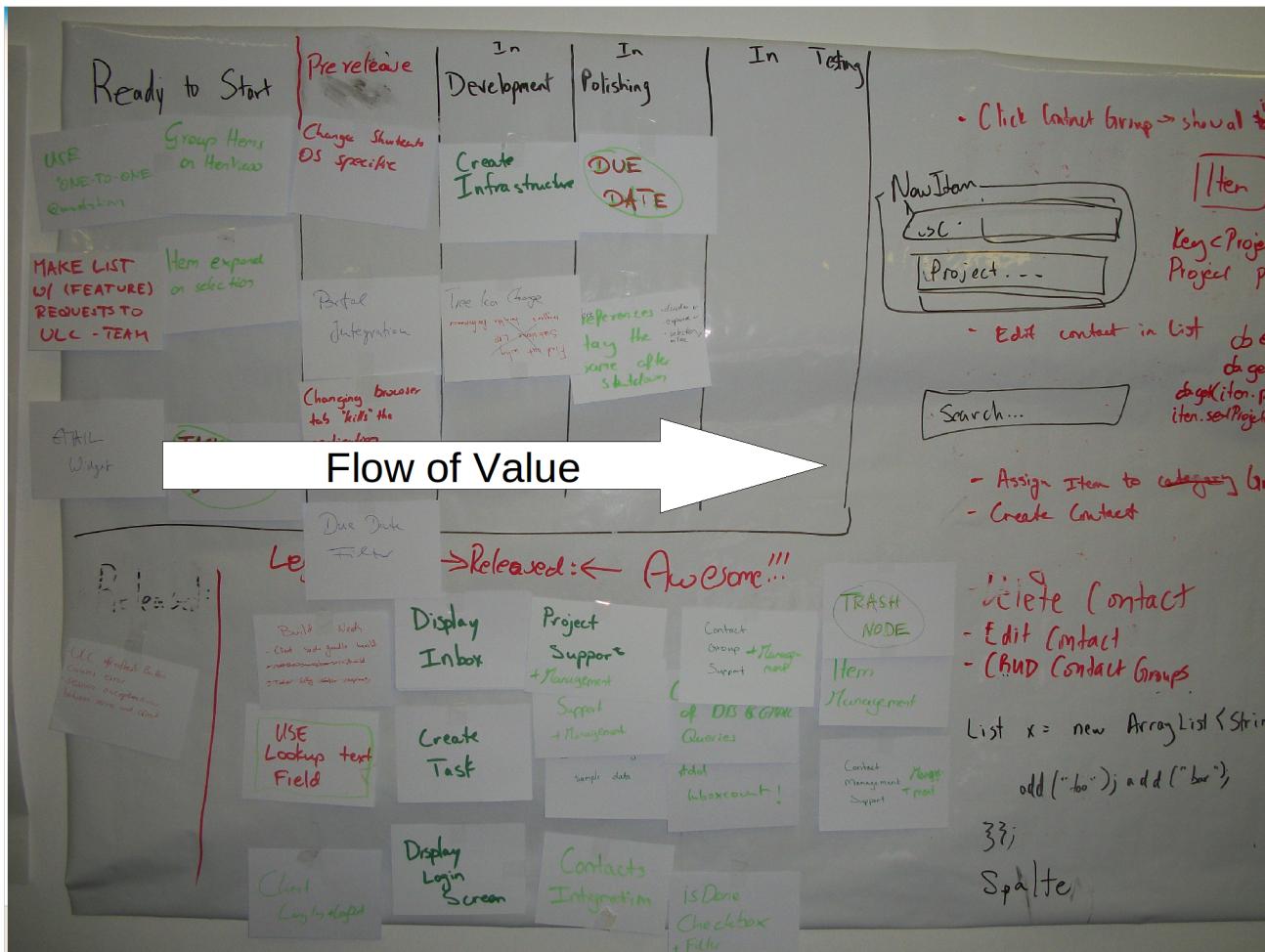
Lean Principles

- Eliminate Waste
- Build Quality In
- Create Knowledge
- Defer Commitment
- **Deliver Fast**
- Respect People
- Optimize the Whole



Spoiled Sashimi and Deployment Charades







Queuing Theory

Delivery becomes slower as...

- variability increases
- batch size increases
- utilization increases



Little's Law and Throughput

Inventory = Arrival Rate × Flow Time

5 People in Ramp Jam (Inventory)

1 Minutes Per Person (Flow Time)

Wait is 5 minutes (Arrival Rate)

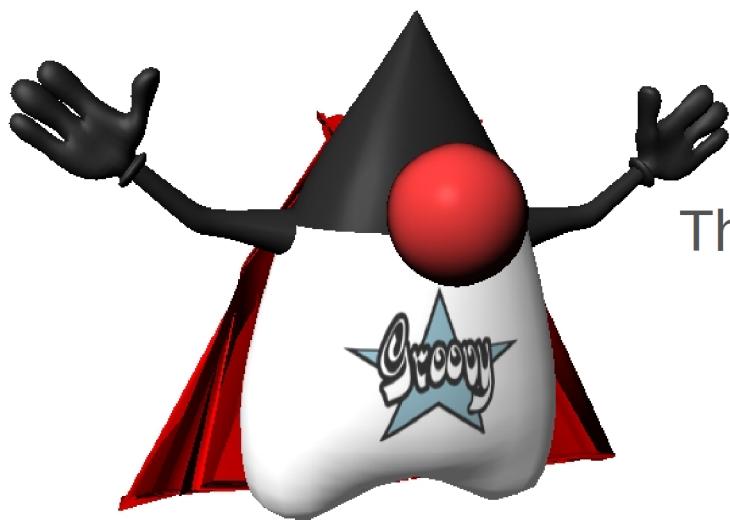
Little's Law and Throughput

You will move user stories to done faster by
reducing the amount of stories in-process

(a simplification)

Little's Law and Throughput

work in small batches
create slack time
reduce variability



The Small Batch Language



Are your automated tests
helping you go faster?

Platform Services

Gold Card

Was sind Gold Cards?

- Zeitfenster
- Gold Cards
- Der SW Impact

Wie werden Gold Cards benutzt?

- Jeder SW hat eine Gold Card
- Eine Gold Card ist ein Dokument
- Eine Gold Card ist ein Dokument
- Die Resultate werden verglichen

FRANK

Profile	Profile	Java Spring	4. Sprint	5. Sprint
X				
			14.07.10 – 27.07.10	28.07.10 – 10.08.10
6. Sprint	7. Sprint	Java EE		
11.08.10 – 24.08.10	25.08.10 – 07.09.10			
10. Sprint	11. Sprint	12. Sprint	13. Sprint	14. Sprint
06.10.10 – 19.10.10	20.10.10 – 02.11.10	03.11.10 – 16.11.10	17.11.10 – 30.11.10	01.12.10 – 14.12.10
15. Sprint	16. Sprint	17. Sprint	18. Sprint	
15.12.10 – 28.12.10	29.12.10 – 11.01.11	12.01.11 – 25.01.11	26.01.11 – 08.02.11	
19. Sprint	20. Sprint	21. Sprint	22. Sprint	23. Sprint

Thomas

1. Sprint	2. Sprint	3. Sprint	4. Sprint	5. Sprint
JAXB		Sessions		
02.06.10 – 15.06.10		30.06.10 – 13.07.10		28.07.10 – 10.08.10
6. Sprint	7. Sprint	8. Sprint	9. Sprint	
11.08.10 – 24.08.10	25.08.10 – 07.09.10		22.09.10 – 05.10.10	
10. Sprint	11. Sprint	12. Sprint	13. Sprint	14. Sprint
06.10.10 – 19.10.10	20.10.10 – 02.11.10	03.11.10 – 16.11.10	17.11.10 – 30.11.10	01.12.10 – 14.12.10
15. Sprint	16. Sprint	17. Sprint	18. Sprint	
15.12.10 – 28.12.10	29.12.10 – 11.01.11	12.01.11 – 25.01.11	26.01.11 – 08.02.11	
19. Sprint	20. Sprint	21. Sprint	22. Sprint	23. Sprint

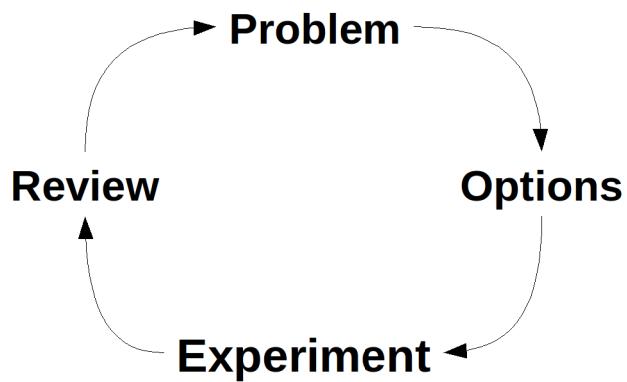
Alex

1. Sprint	2. Sprint	3. Sprint	4. Sprint	5. Sprint
Android	JUNIT Tests	SOAP UI		
06.10.10 – 19.10.10			14.07.10 – 27.07.10	28.07.10 – 10.08.10
6. Sprint	7. Sprint	8. Sprint	9. Sprint	
11.08.10 – 24.08.10	25.08.10 – 07.09.10		22.09.10 – 05.10.10	
10. Sprint	11. Sprint	12. Sprint	13. Sprint	14. Sprint
06.10.10 – 19.10.10	20.10.10 – 02.11.10	03.11.10 – 16.11.10	17.11.10 – 30.11.10	01.12.10 – 14.12.10
15. Sprint	16. Sprint	17. Sprint	18. Sprint	
15.12.10 – 28.12.10	29.12.10 – 11.01.11	12.01.11 – 25.01.11	26.01.11 – 08.02.11	
19. Sprint	20. Sprint	21. Sprint	22. Sprint	23. Sprint

Sebastian

1. Sprint	2. Sprint	3. Sprint	4. Sprint	5. Sprint
Java FX	Fitnesse			
11.08.10 – 24.08.10				
6. Sprint	7. Sprint	8. Sprint	9. Sprint	
11.08.10 – 24.08.10	25.08.10 – 07.09.10		22.09.10 – 05.10.10	
10. Sprint	11. Sprint	12. Sprint	13. Sprint	14. Sprint
06.10.10 – 19.10.10	20.10.10 – 02.11.10	03.11.10 – 16.11.10	17.11.10 – 30.11.10	01.12.10 – 14.12.10
15. Sprint	16. Sprint	17. Sprint	18. Sprint	
15.12.10 – 28.12.10	29.12.10 – 11.01.11	12.01.11 – 25.01.11	26.01.11 – 08.02.11	
19. Sprint	20. Sprint	21. Sprint	22. Sprint	23. Sprint

Recipes for Going Faster



From “Agile Coaching”
(a very good book)



Recipe for Disaster:

increase productivity in a non-bottleneck area

Next Steps: Profit and Loss Sheet

What has the greatest cost?

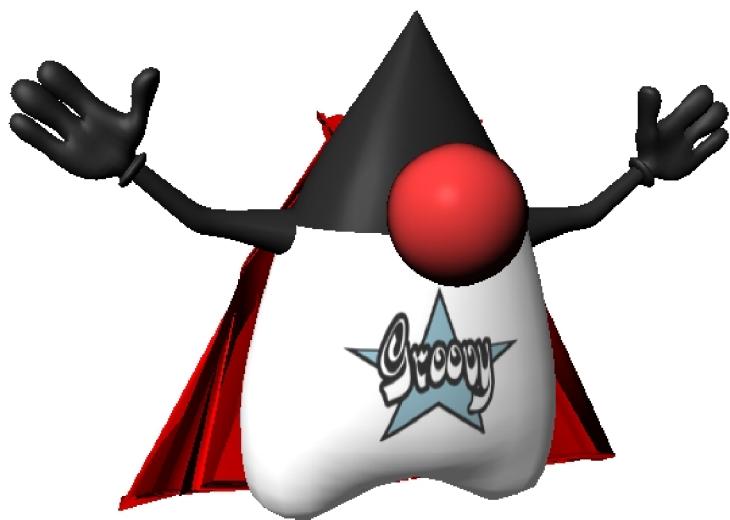
- delivery four week later
- refactoring for 40 hours
- paying a \$100,00 license fee
- skipping user interface design
- ignoring risk of team member leaving

Lean Principles

- Eliminate Waste
- Build Quality In
- Create Knowledge
- Defer Commitment
- Deliver Fast
- Respect People
- Optimize the Whole

... some anti patterns

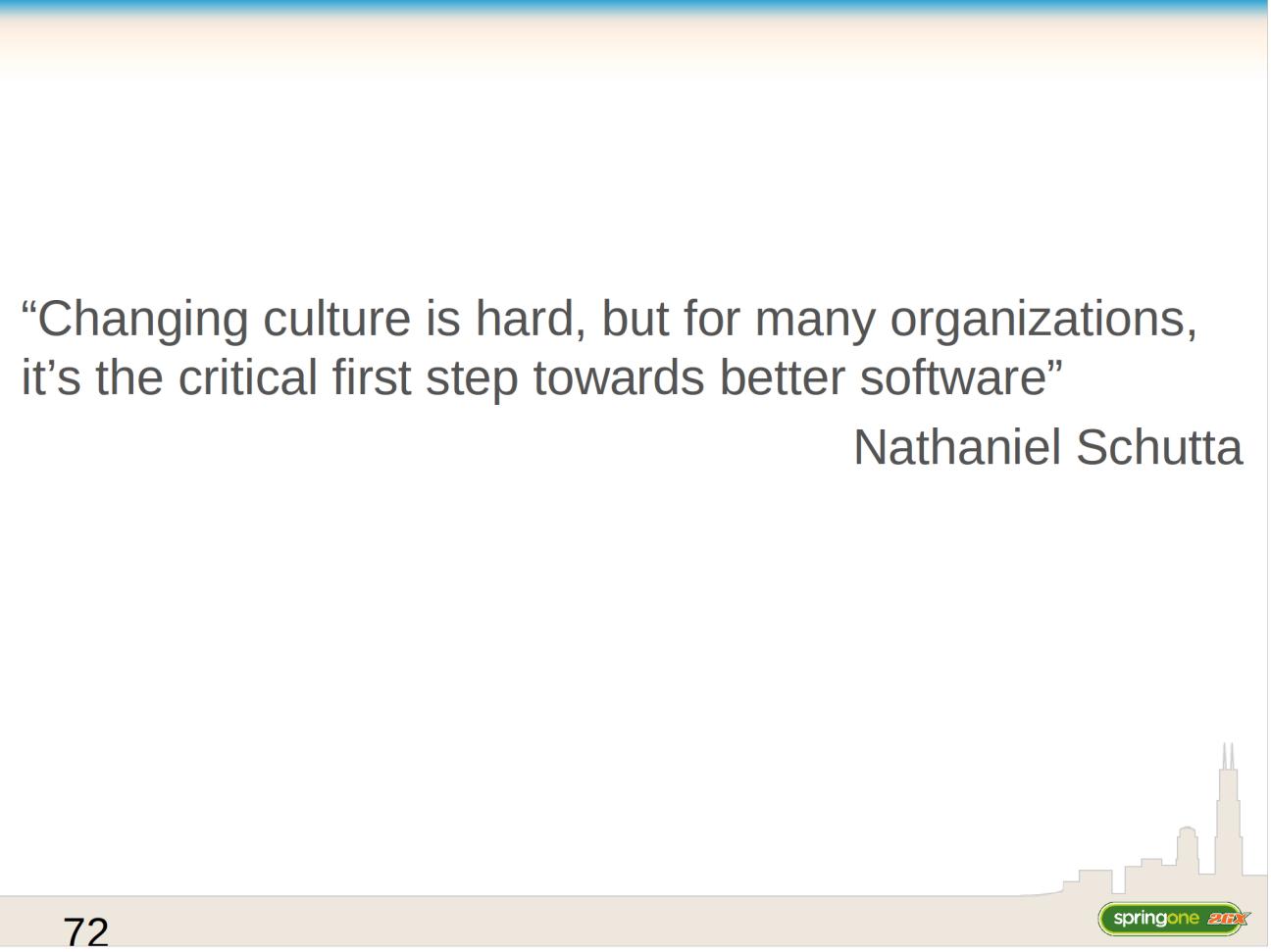




we're all in this
together people



springone 2GX



“Changing culture is hard, but for many organizations,
it’s the critical first step towards better software”

Nathaniel Schutta

Agile Adoption Stage 0

behave territorially
are inflexible
grow uncomfortable with uncertainty
treat developers as a commodity
believe development is a linear process

(you're not ready)

Backdoor Groovy

SOAP UI and Web Services
@Grab and Groovy with Operations



springone 2GX

Demo





“You're free to
use Groovy for
personal scripting”



springone 

Lean Principles

- Eliminate Waste
- Build Quality In
- Create Knowledge
- Defer Commitment
- Deliver Fast
- Respect People
- Optimize the Whole

Documentation Breeds in Captivity

Which is the canonical source?

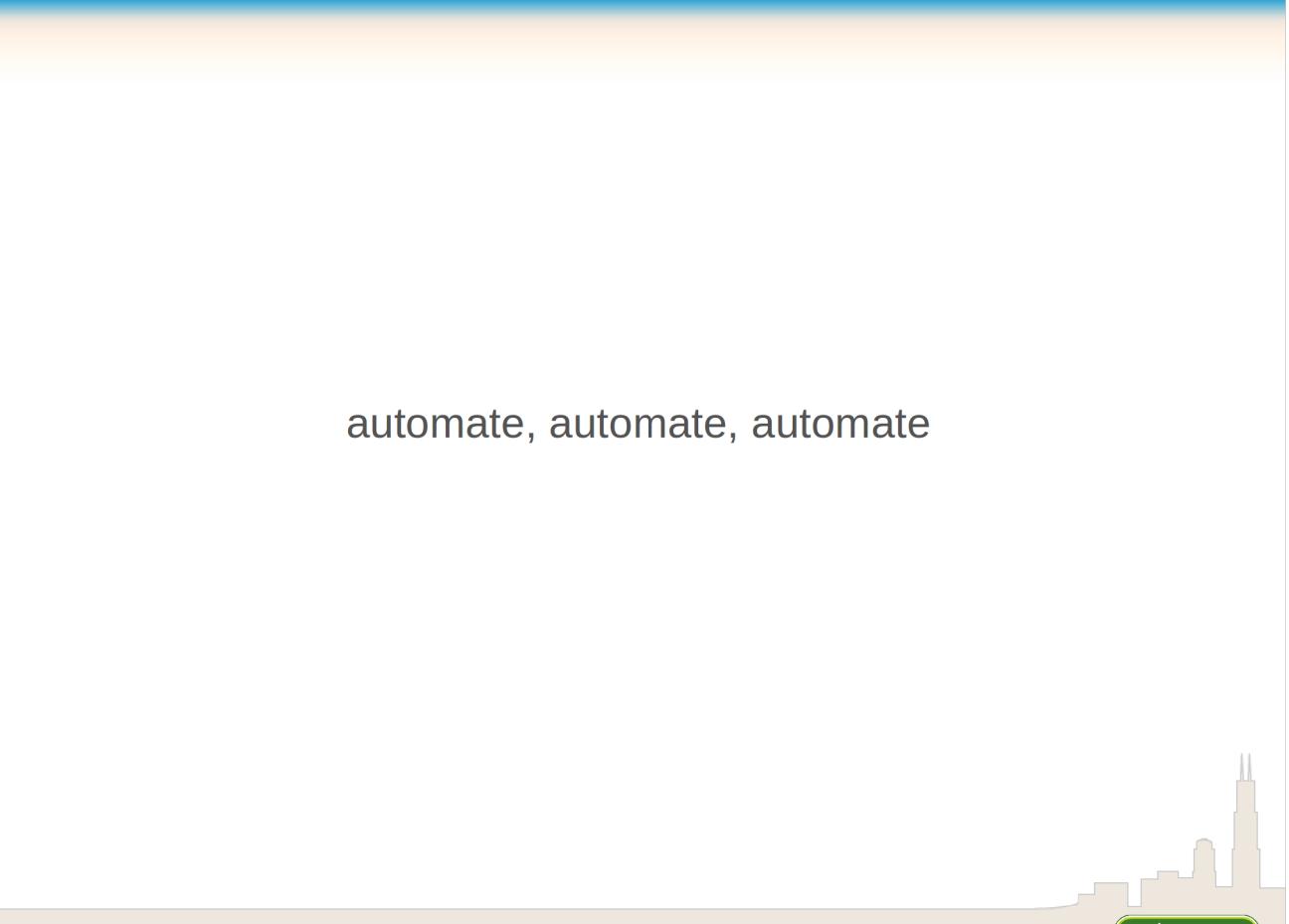
Requirements

User Stories

Functional Tests

Test Plans

Demo



automate, automate, automate

'automate' * 3

better?

Gradle

a better way to build

A platform for automation
(and covered better elsewhere)

Lean Principles

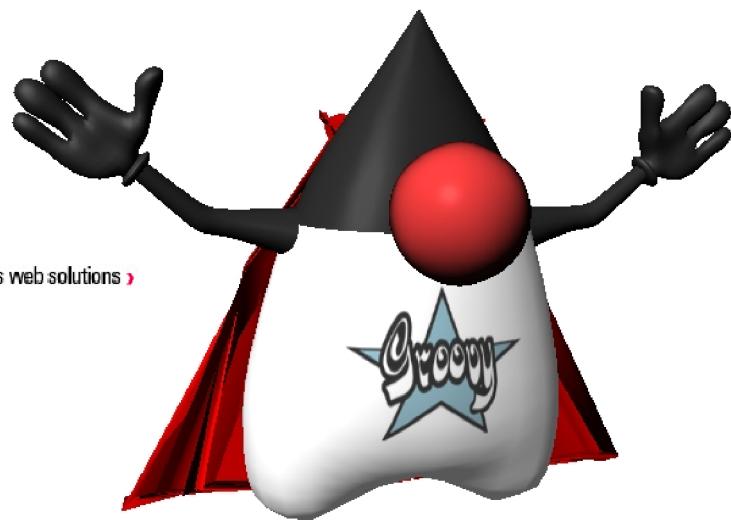
- Eliminate Waste
- Build Quality In
- Create Knowledge
- Defer Commitment
- Deliver Fast
- Respect People
- Optimize the Whole

“Be a catalyst for change”
Pragmatic Programmer Tip 5

Q&A

canoo

› your provider for business web solutions ›



Griffon, Grails, Groovy, and Agile Consulting
info@canoo.com or hamlet.darcy@canoo.com