



Chicago, October 19 - 22,
2010

Slimmed Down Software A Lean, Groovy Approach

Hamlet D'Arcy
Canoo Engineering AG
@HamletDRC





canoo
your provider for business web solutions >





<http://nofluffjuststuff.com/>



Slimmed Down Software – A Lean, Groovy Approach Part 1 – Eliminate Waste

This article originally appeared in the April 2010 edition of [GroovyMag](#), the Groovy and Grails magazine. Parts 1 and 3 are currently available for download from the magazine's site, and more will come each month. Enjoy!

The Groovy Programming Language advertises itself as an "agile and dynamic language for the JVM™", but what does this mean exactly? This series of articles explains *Lean Software Development*, and shows how your choice of programming language can make your entire process remain nimble and adaptive. Each month will cover one of the seven *Lean Software Development* principles and explain how Groovy and the associated ecosystem help eliminate waste, defer commitment, and build quality into your product.

About This Series

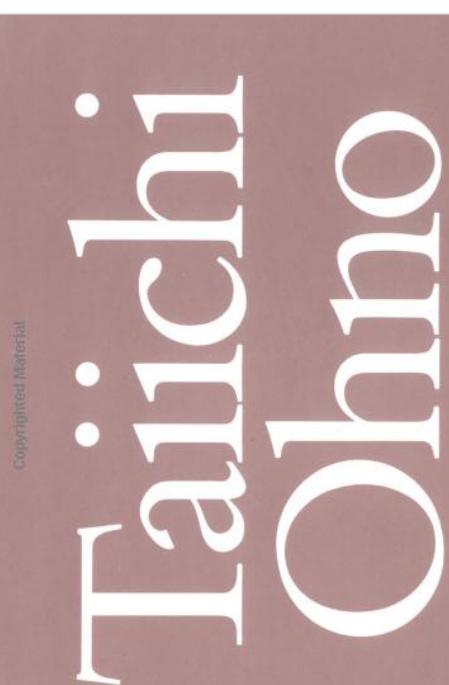
Most of your agile team will spend the majority of their day working with a programming language, making it one of the most important tools to optimize in your project's lifecycle. Yet many organizations are reluctant to experiment with newer languages and frameworks, instead preferring to stick with what works in one domain or another.

<http://groovymag.com/blog>

GROOVY MAG
for groovy and grails developers



1988 - Japan



Copyrighted Material

2003 - Minnesota



The Agile Software Development Series

Cockburn • Highsmith
Series Editors

- Adapting agile practices to your development organization
- Uncovering and eradicating waste throughout the software development lifecycle
- Practical techniques for every development manager, project manager, and technical leader

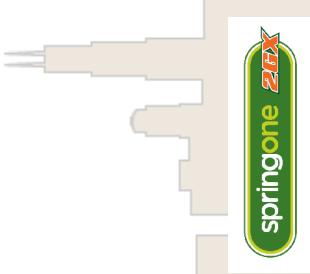
Forewords by
Jim Highsmith
and **Ken Schwaber**

Mary Poppendieck
Tom Poppendieck

Copyrighted Material

Lean Principles

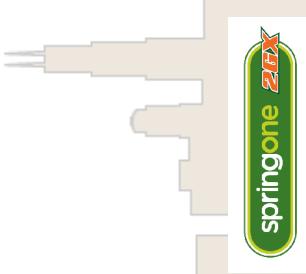
- Eliminate Waste
- Build Quality In
- Create Knowledge
- Defer Commitment
- Deliver Fast
- Respect People
- Optimize the Whole



Lean Principles

XP Values

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



Lean Principles

XP Values

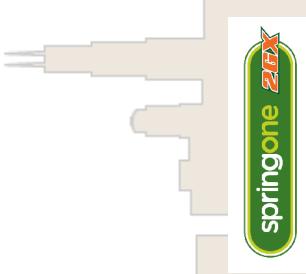
- Eliminate Waste
- Build Quality In
- Create Knowledge
- Defer Commitment
- Deliver Fast
- Respect People
- Optimize the Whole

XP Principles

- Communication
- Simplicity
- Feedback
- Courage
- Respect
- Feedback

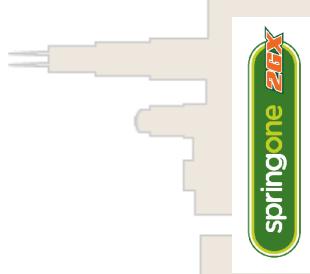
Lean Principles

- 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



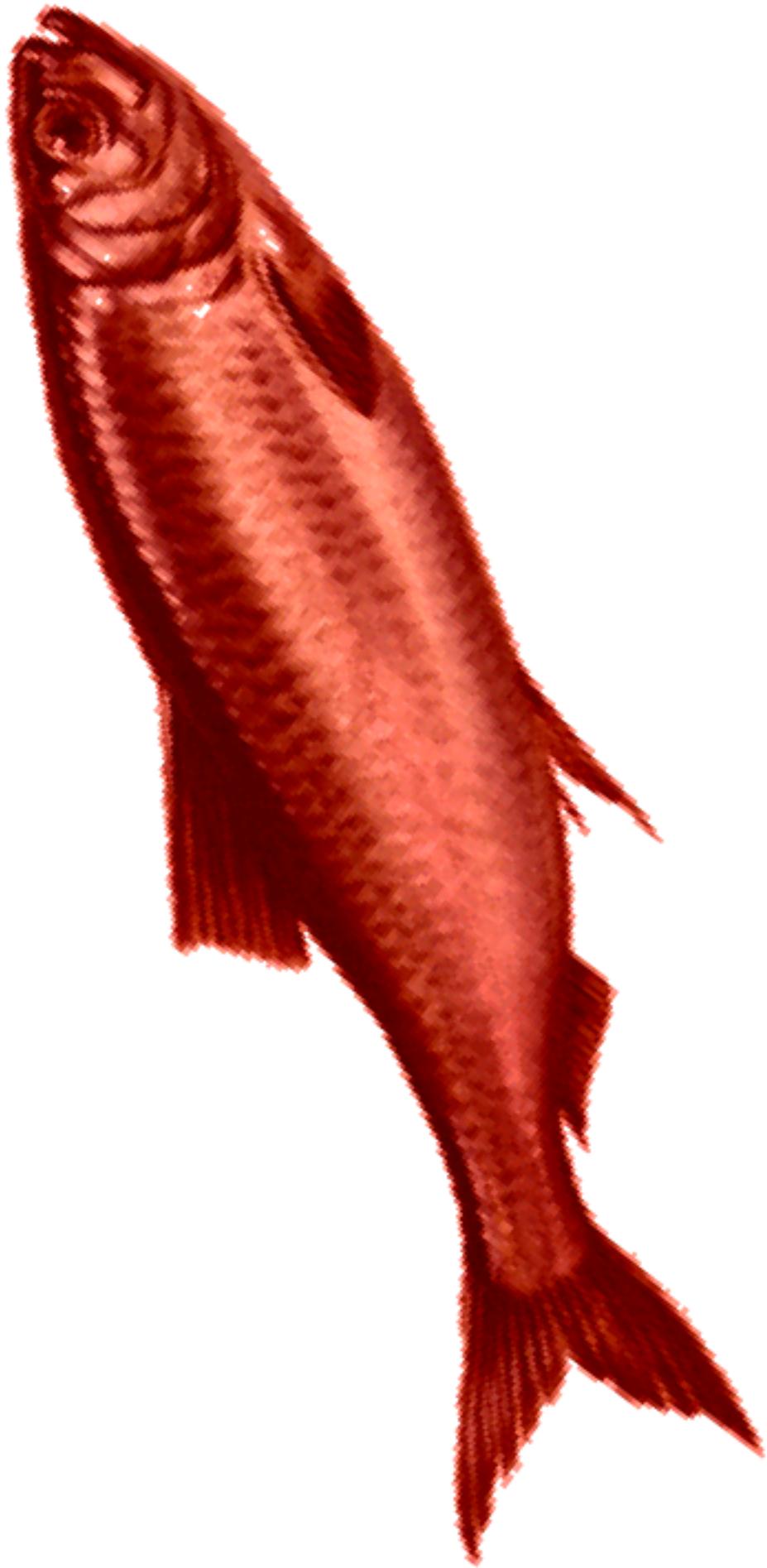
Eliminate Waste?

```
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?



Phase 1

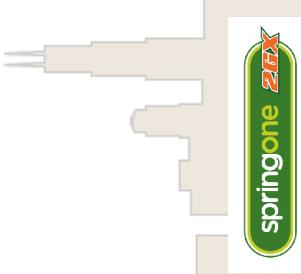
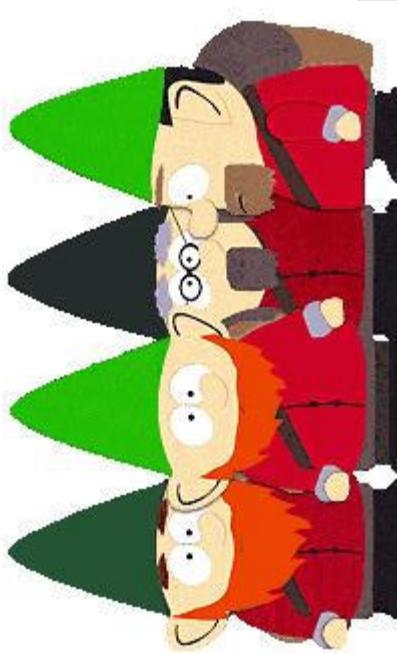
Phase 2

Phase 3

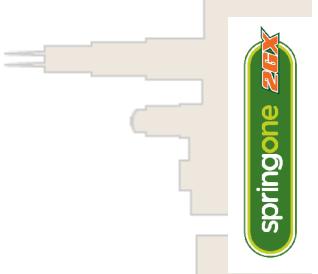
stop doing
things that
aren't worth
the effort

do the same
things with
less effort

Profit!



Membra Them?!

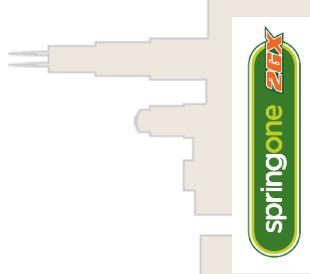


Deja Vu?

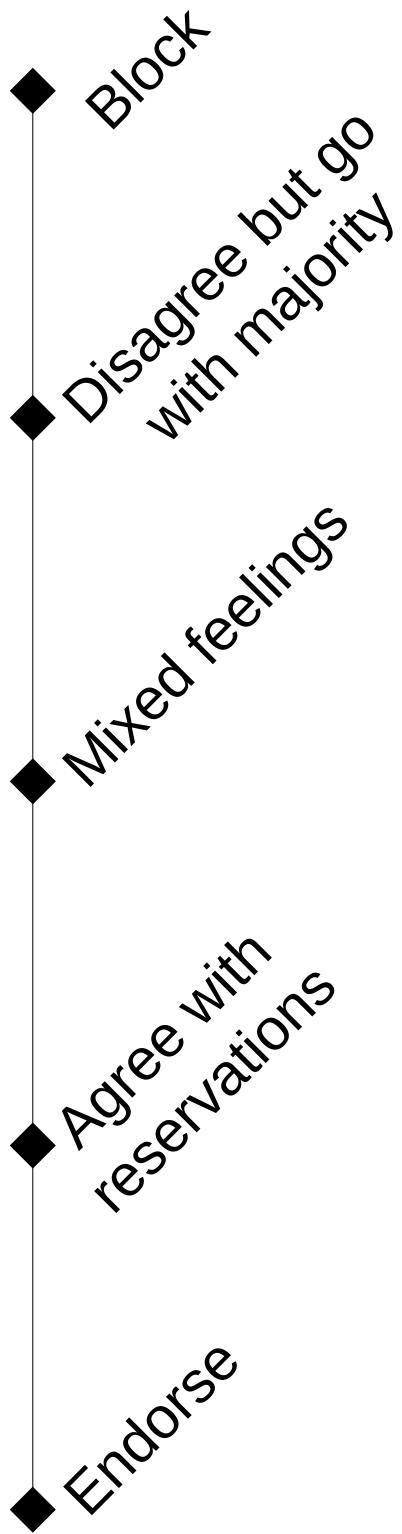


How to Make a Decision

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



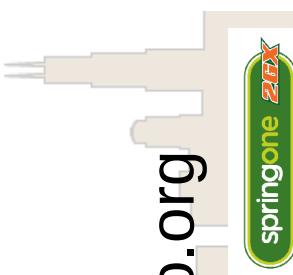
From “Agile Coaching”



Write Only Process Improvement

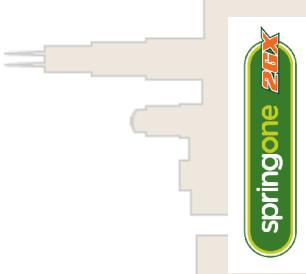
Working software over comprehensive documentation
as long as that software is comprehensively documented.

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



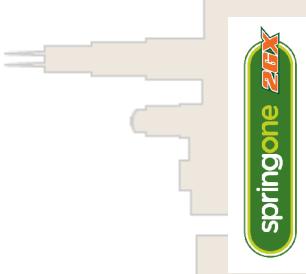
Recognizing Waste

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

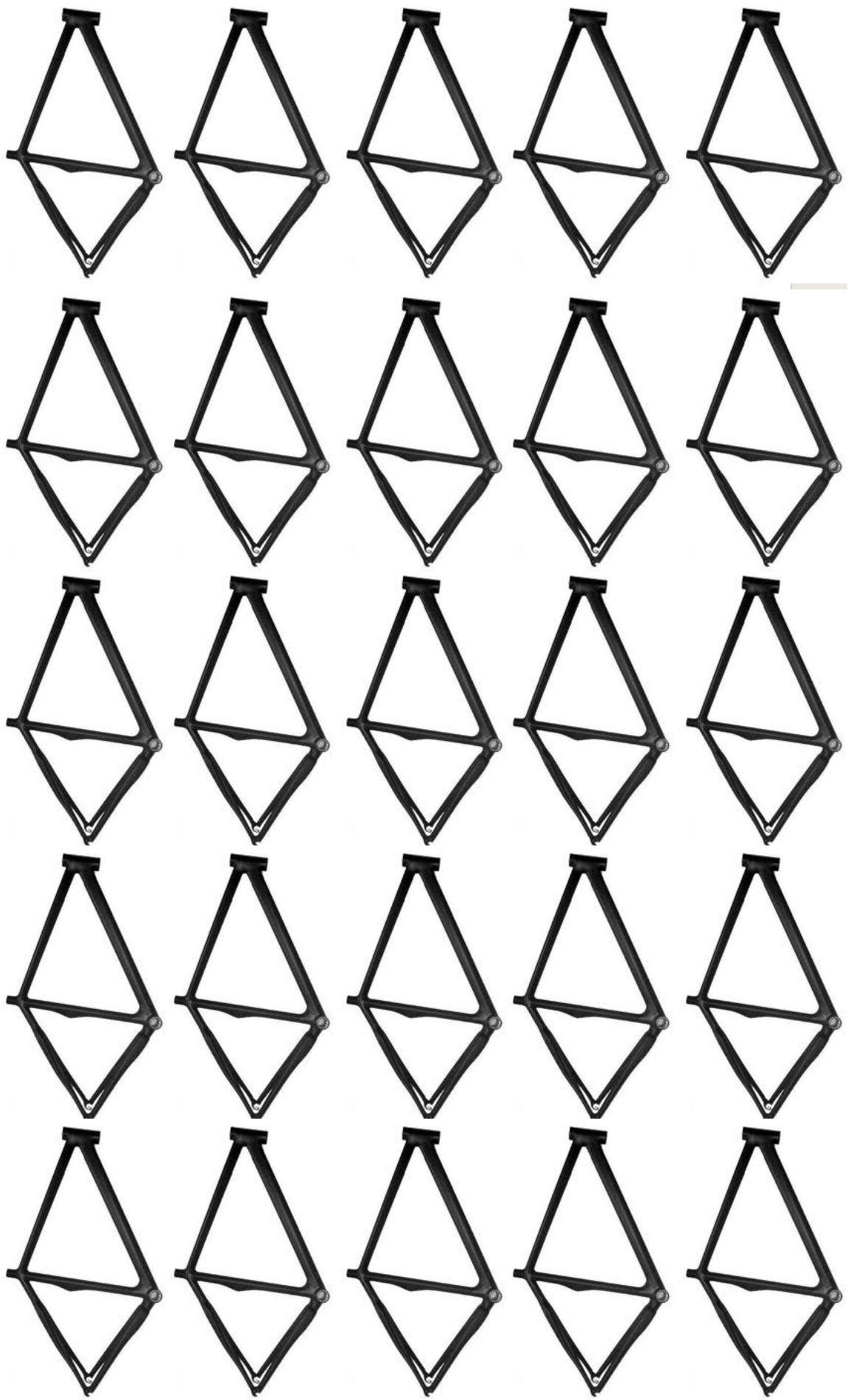


Recognizing Waste

- 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

```

<binding name="TestsoapBinding" type="tns:TestsoapPortType">
  <soap:binding transport="http://schemas.xmlsoap.org/http" />
  <operation name="Multiply">
    <soap:operation style="rpc" soapAction="http://seapinerop.org/Multiply" />
    <input>
      <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded" />
      <namespace>http://seapinerop.org</namespace>
      <name>multiply</name>
      <type>tns:multiply</type>
    </input>
    <output>
      <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded" />
      <namespace>http://seapinerop.org</namespace>
      <name>multiplyResponse</name>
      <type>tns:multiplyResponse</type>
    </output>
  </operation>
  <operation name="Add">
    <soap:operation style="document" soapAction="http://seapinerop.org/Add" />
    <input>
      <soap:body use="literal" />
      <namespace>http://seapinerop.org</namespace>
      <name>add</name>
      <type>tns:add</type>
    </input>
    <output>
      <soap:body use="literal" />
      <namespace>http://seapinerop.org</namespace>
      <name>addResponse</name>
      <type>tns:addResponse</type>
    </output>
  </operation>
  <operation name="GetTime">
    <soap:operation style="rpc" soapAction="http://seapinerop.org/GetTime" />
    <input>
      <soap:body use="encoded" />
      <namespace>http://seapinerop.org</namespace>
      <name>GetTimeRequest</name>
      <type>tns:GetTimeRequest</type>
    </input>
    <output>
      <soap:body use="encoded" />
      <namespace>http://seapinerop.org</namespace>
      <name>GetTimeResponse</name>
      <type>tns:GetTimeResponse</type>
    </output>
  </operation>
</binding>
<service name="Testsoap1DocAndRpcService">
  <port name="Testsoap1Binding" binding="tns:TestsoapBinding">
    <soap:address location="http://localhost/services/tns:interopDocAndRpc" />
  </port>
</service>
</definitions>

```

```

<binding name="TestSoapBinding" type="tns:TestSoapPartType" >
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <operation name="Multiply" />
    <soap:operation style="http://schemas.xmlsoap.org/Multiply" />
    <input>
      <soap:encodingStyle "http://schemas.xmlsoap.org/soap/encoding/" use="encoded" />
      <name>ns2:ns2</name>
      <namespace>http://sapintriper.org</namespace>
      <input>
        <outbound>
          <soap:header encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded" />
          <name>ns2:ns2</name>
          <namespace>http://sapintriper.org</namespace>
          <group>ns2:ns2</group>
          <order index="1" />
        </outbound>
        <operation style="document" soapAction="http://sapintriper.org/Add" />
        <requestBody use="literal" />
        <responseBody use="literal" />
      </input>
      <outbound>
        <soap:header encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded" />
        <name>ns2:ns2</name>
        <namespace>http://sapintriper.org</namespace>
        <group>ns2:ns2</group>
        <order index="1" />
      </outbound>
    </input>
  </operation>
</binding>
<definitions>
  <service name="TestService" binding="tns:TestSoapBinding">
    <soap:address location="http://localhost/services/mstintert1DocAndRfcAndRpc" />
  </service>
</definitions>

```

```

<binding name="TestSoapBinding" type="tns:TestSoapPortType">
    <soap11 binding transport="http://schemas.xmlsoap.org/soap/http" />
    <creation binding name="MultiOp" />
    <operation name="multiOp" style="rpc" soapaction="http://soainterop.org/multiOp" />
        <input>
            <soap11 encodingStyle="http://schemas.xmlsoap.org/soap/encoding" use="encoded" />
            <name>param</name>
            <type>http://soainterop.org/param</type>
        </input>
        <output>
            <soap11 encodingStyle="http://schemas.xmlsoap.org/soap/encoding" use="encoded" />
            <name>param</name>
            <type>http://soainterop.org/param</type>
        </output>
    </operation>
    <operation name="Add" />
        <input>
            <soap operation style="document" soapaction="http://soainterop.org/Add" />
            <name>param</name>
            <type>http://soainterop.org/param</type>
        </input>
        <output>
            <soap11 encodingStyle="http://schemas.xmlsoap.org/soap/encoding" use="encoded" />
            <name>param</name>
            <type>http://soainterop.org/param</type>
        </output>
    </operation>
    <operation name="Bind" />
        <input>
            <soap11 encodingStyle="http://schemas.xmlsoap.org/soap/encoding" use="encoded" />
            <name>param</name>
            <type>http://soainterop.org/param</type>
        </input>
        <output>
            <soap11 encodingStyle="http://schemas.xmlsoap.org/soap/encoding" use="encoded" />
            <name>param</name>
            <type>http://soainterop.org/param</type>
        </output>
    </operation>

```

```

<binding name="TestSoapBinding" type="tns:TestSoapType">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="MultiOp" />
    <operation style="rpc" soapAction="http://soapsinterop.org/MultiOp" />
    <input />
    <output />
    <soapBody encodingStyle="http://schemas.xmlsoap.org/soap/encoding" />
    <ns:document><ns:multiOp></ns:multiOp></ns:document>
    <ns:multiOp><ns:multiOpResponse></ns:multiOpResponse></ns:multiOp></ns:document>
    <ns:multiOpResponse><ns:multiOpResult></ns:multiOpResult></ns:multiOpResponse></ns:document>
</binding>
<binding name="TestSoapBinding" type="tns:TestSoapType">
    <soap:binding transport="http://soapsinterop.org/Add" />
    <operation name="Add" />
    <operation style="document" soapAction="http://soapsinterop.org/Add" />
    <input />
    <output />
    <soapBody use="literal" />
    <ns:document><ns:add></ns:add></ns:document>
    <ns:add><ns:addResponse></ns:addResponse></ns:add></ns:document>
    <ns:addResponse><ns:addResult></ns:addResult></ns:addResponse></ns:document>
</binding>
<binding name="TestSoap" type="tns:TestSoap" />
<service name="MSInterOpAndRPCService" />
<port binding="TestSoap" name="TestSoapPort" />
<address location="http://localhost/services/mstinteropandrpc" />
<service />

```

Code without Tests is Half Code



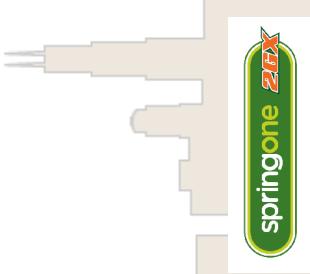
Demo

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

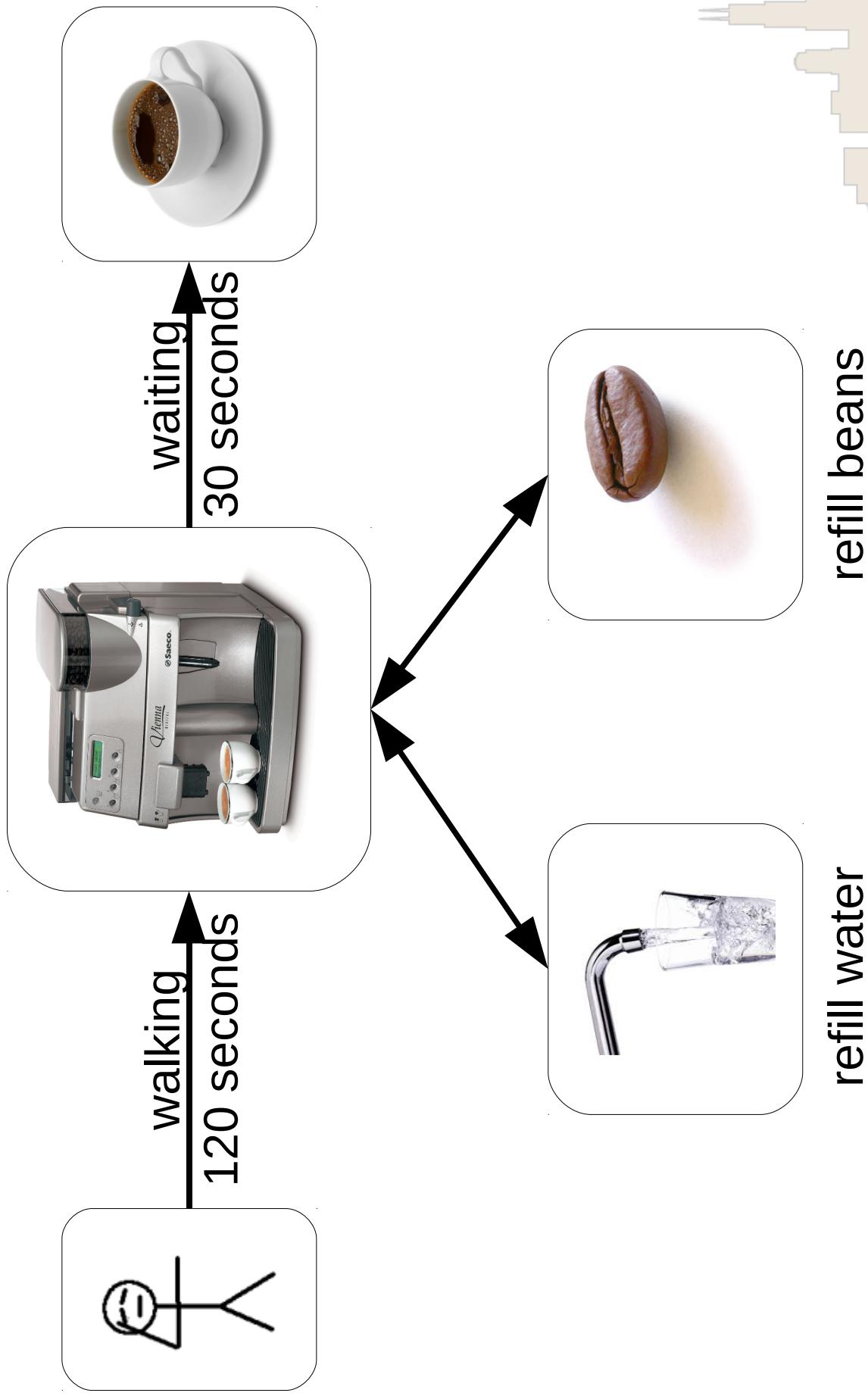


There is a big difference between
eliminating waste and not creating it in the
first place.

Alan Shalloway



Next Steps: Value Stream Mapping

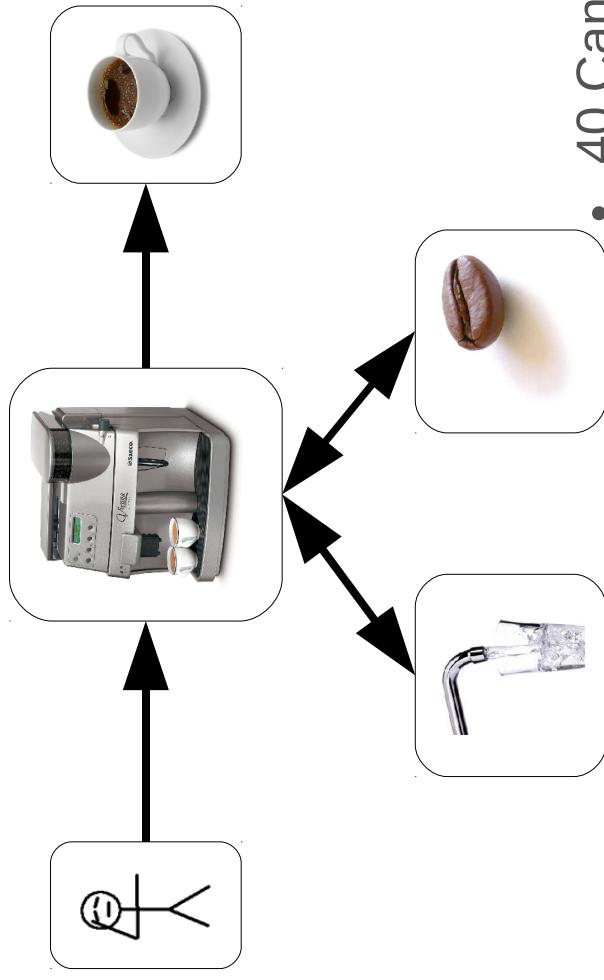


120 / 10
refill water

refill beans

120 / 10
refill water

Next Steps: Value Stream Mapping



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

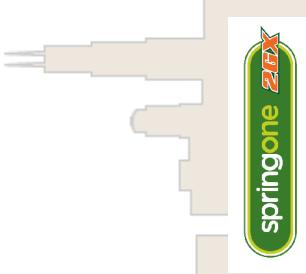


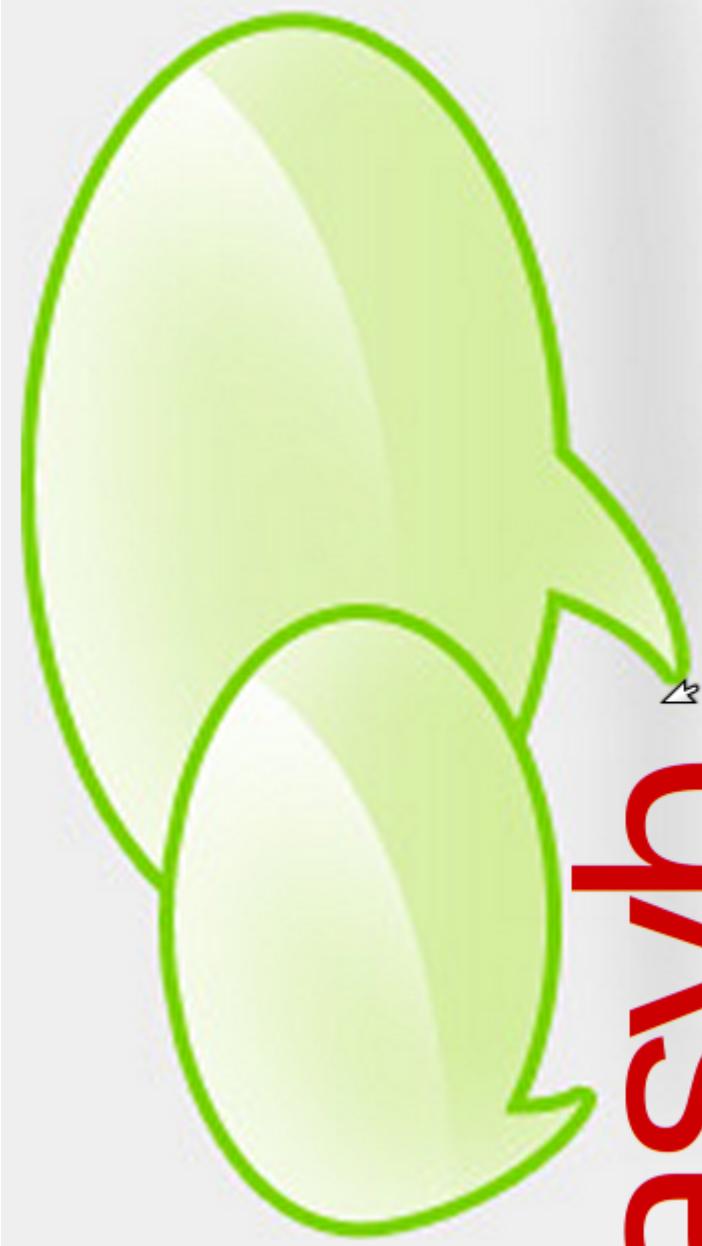
The screenshot shows a web browser displaying the Agile Zone website. The main page header features the 'AGILE ZONE' logo with the subtitle 'Software Methodologies for Development Managers'. Below the header, there are navigation links for Home, Zones, Library, Refcardz, Forums, Links, TQuestions, Snippets, and DZone. The main content area is titled 'The Seven Wastes of Software Development' and is submitted by Matt Stone on Wed, 2010-08-11, 12:00am. The article discusses the 'Seven Wastes' concept, originally from lean manufacturing, applied to software development. It includes sections on 'Getting Groovy with Flex', '7 Metrics of Software Development', and 'Popular at DZone'. The sidebar on the right contains links to other articles and news items related to software development.



Lean Principles

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





easyb

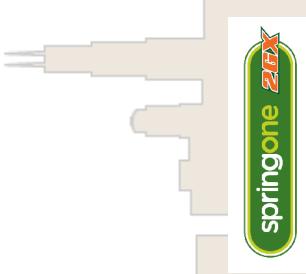
-- bdd in java can't get any easier

Demo

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



Really?!: Behavior Driven Development



Next Steps:

Zero Defects

Stop the Line

~~BITS Agile 1.0~~

~~Zero Defects - Stop the line~~

Daily Standup at 9:00 am
2.1 2.2 2.3 2.4

2 week iterations

Demo Release Required **EVERY FRIDAY**

↳ email customer Release Notes + time remaining in budget

Iteration Retropective = with Customer
↳ rotating facilitator
= separate from standup
→ NEW FORMAT

Story Cards w/ green + yellow
-5 Day Cards. Work on same card. Card Design Sessions
Card Acceptance Criteria
Shelf is Story board

Jira - track demo release progress
↳ USE TO ESTIMATE + REPORT DEFECT LOAD
3rd Party - Get integration help / directions from 3rd Party Vendors

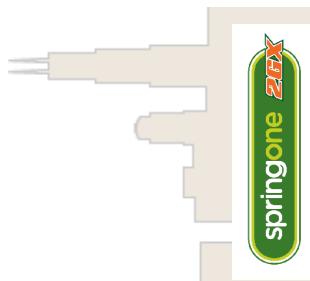
Backlog on Story Cards
↳ Define Stories before putting them in iteration.

Effort + BurnDown in Excel
• Broadcast Agile Successes
on Convo blog + reading group
• Experiment with Pomodoro Technique
CONTINUE

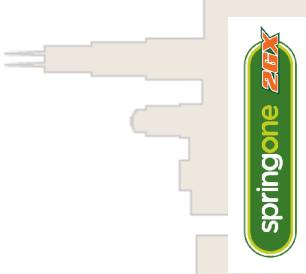


Lean Principles

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

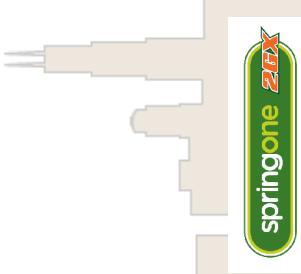


Software Engineer
or
Computer Programmer?



Software Development as Knowledge Acquisition

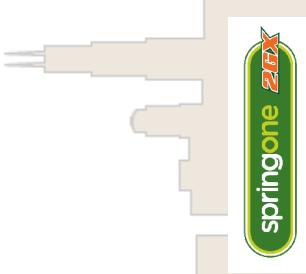
- short, frequent learning cycles
- delayed commitment



Large requirements and specification documents provide a wonderful illusion of certainty

Can tests be Documentation?

Evangelists have oversold
Developers have under-delivered



Demo

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



Unit Test Results: com.canoo.grasp.StoreTest

Designed for use with [JUnit](#) and [Ant](#).

Class com.canoo.grasp.StoreTest

Name	Tests	Errors	Failures	Time(s)	Time Stamp	Host
StoreTest	5	0	0	1.028	2010-04-23T21:33:09	hdarcy-laptop

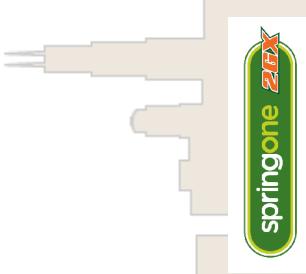
Tests

Name	Status	Type	Time(s)
find must work after save performed	Success		0.462
listener is notified after a save	Success		0.018
listener is not notified twice after two saves	Success		0.029
listener is notified on delete	Success		0.008
listener is not updated after being removed	Success		0.025

[Properties](#)

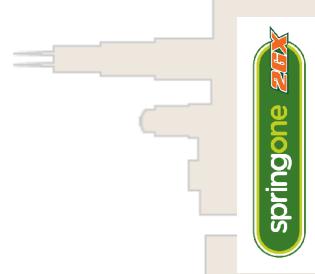
Lean Principles

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



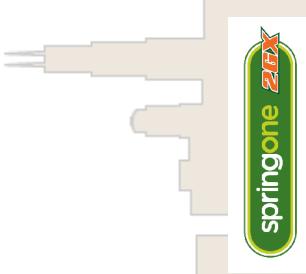
3 Types of Decisions

- a right decision
- a wrong decision
- no decision



3 Types of Decisions

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



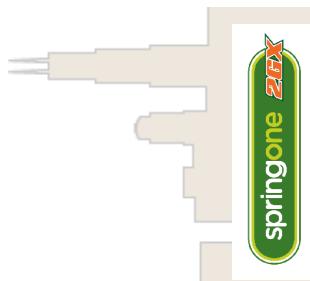
How will you get home tonight?

Last bus

Last train

Taxi

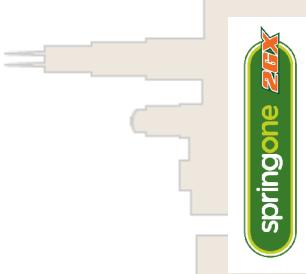
Stay at hotel



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

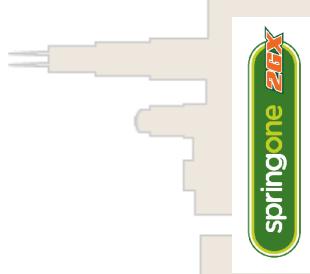


How will you get home tonight?

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



How will you get home tonight?

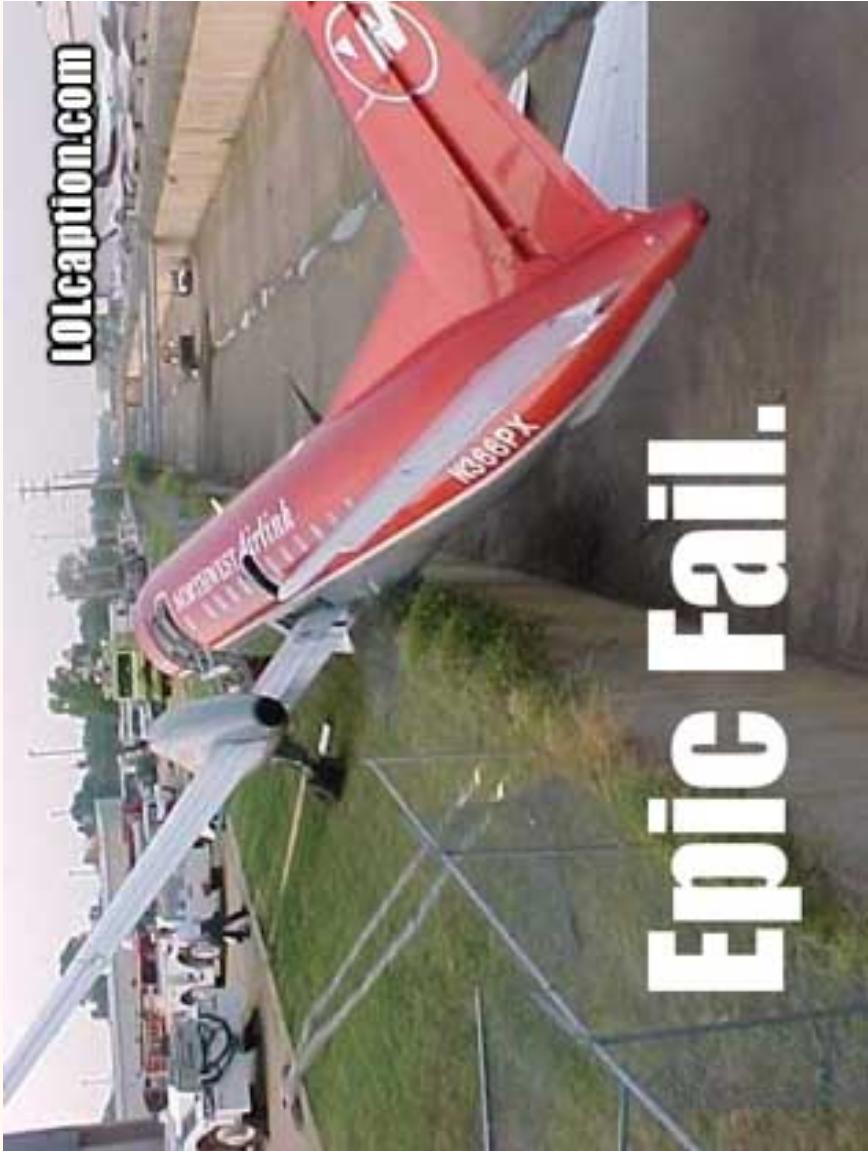
Last bus	\$4.00	11 PM
Last train	\$8.00	12 PM
Taxi	\$25.00	All Night
Stay at hotel	\$99.00	All Night

Options have value

Options expire

Never commit early unless you know why

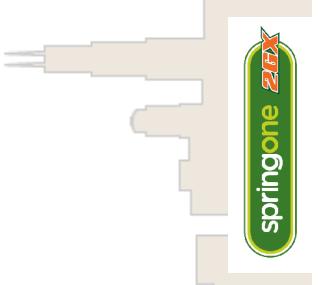




“The last responsible moment is after which execution of an option is too late. The option no longer exists or you’re not able to execute it any more.”

... weekly lesson in
delayed commitment

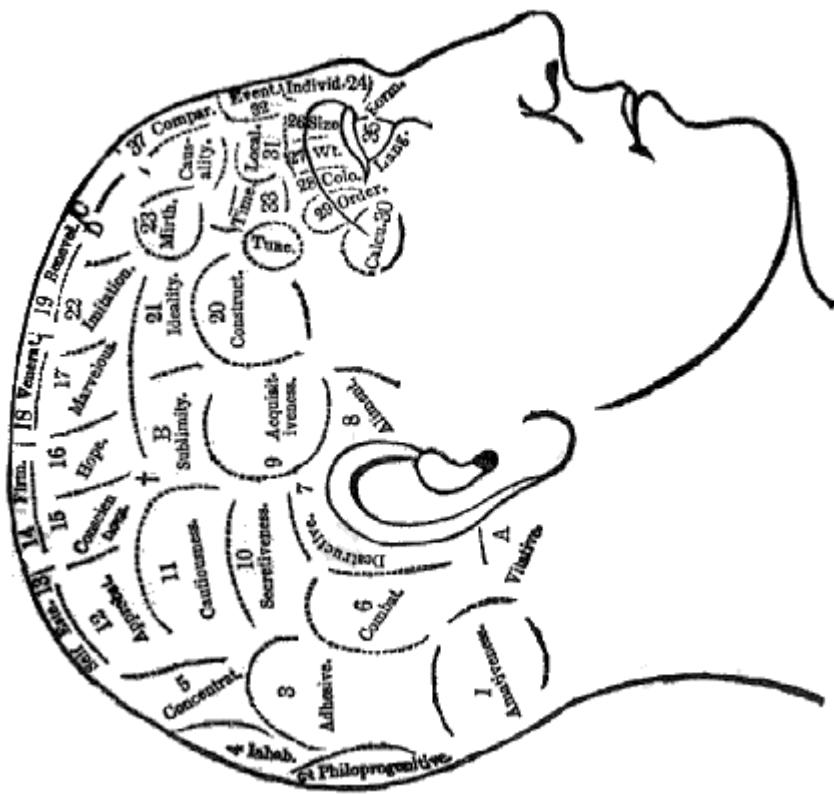




springone ZENX

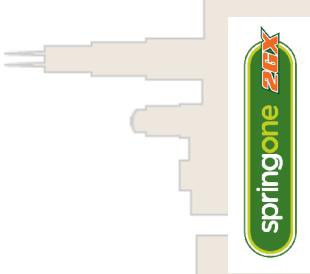
Humans hate uncertainty.

So give them specifics.



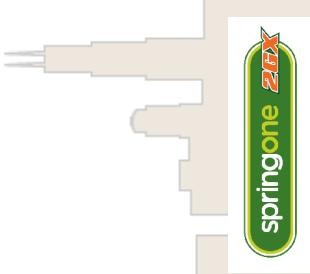
“Very dynamic languages like Lisp, TCL and Smalltalk are often used for prototyping... [One] reason given for these languages being good for prototyping is that they don't require you to pin down decisions early on. JAVA has exactly the opposite property: it forces you to make choices explicitly.”

James Gosling 1996

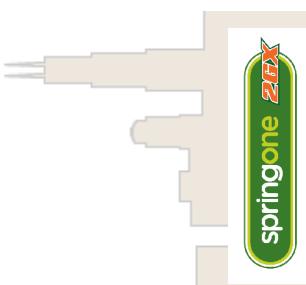


Late Bound and Low Coupled

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



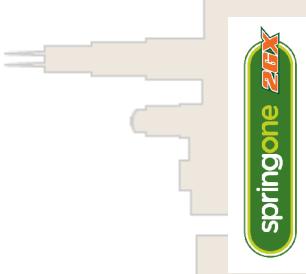
Is the value of an IT system in
the inputs or the outputs?



Groovy for Unit Testing

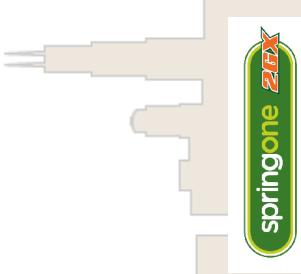
Deferring Commitment

An Undo-able Decision



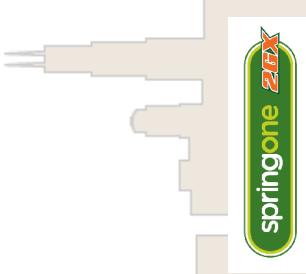
Next Steps: Delayed Commitment Applied

- 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

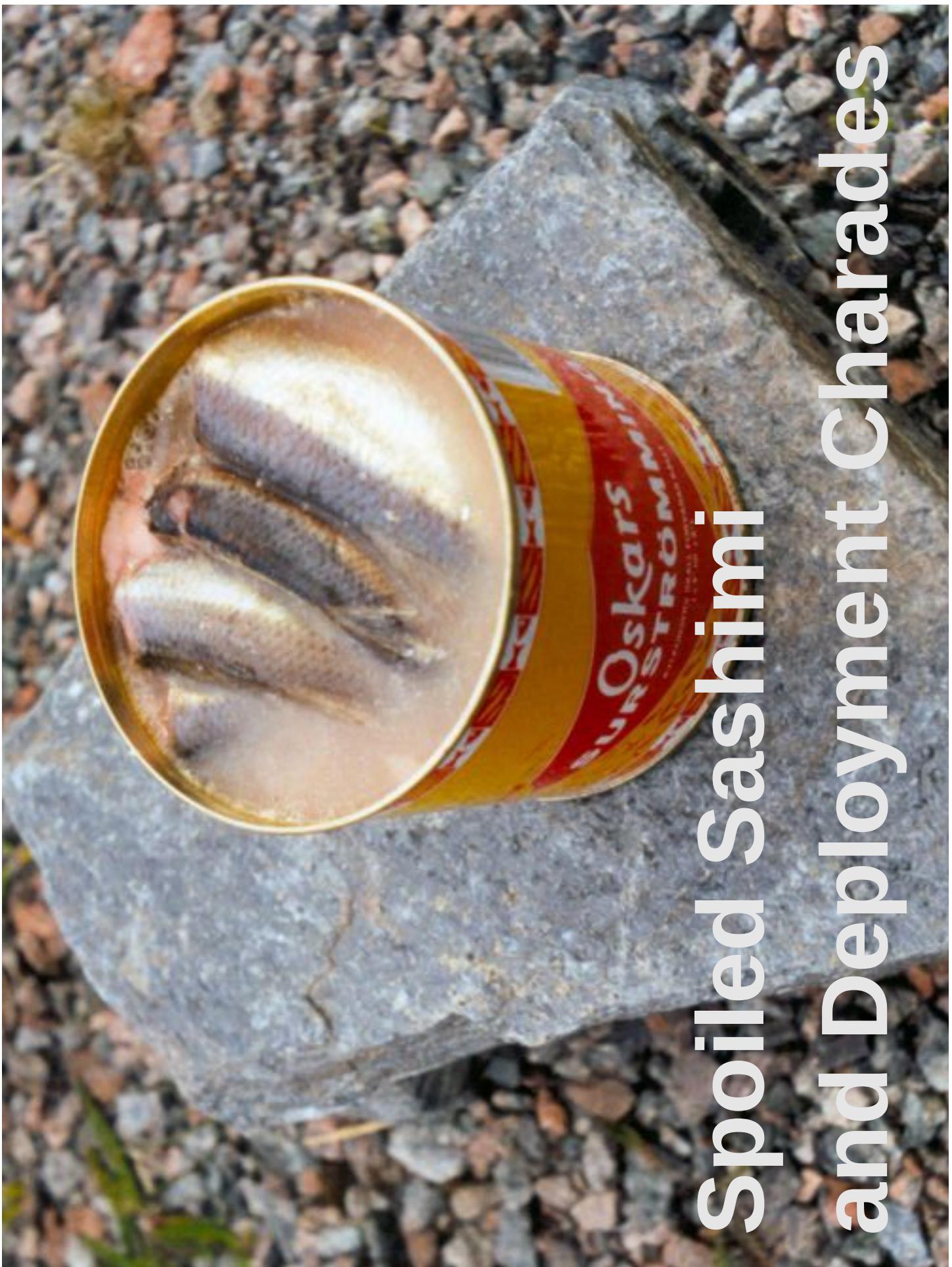


Lean Principles

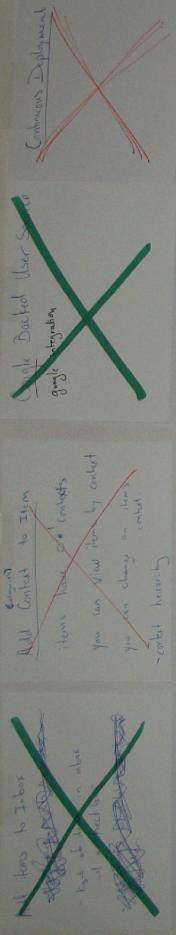
- Eliminate Waste
- Build Quality In
- Create Knowledge
- Defer Commitment
- **Deliver Fast**
- Respect People
- Optimize the Whole



Spoiled Sashimi
and Deployment Charades



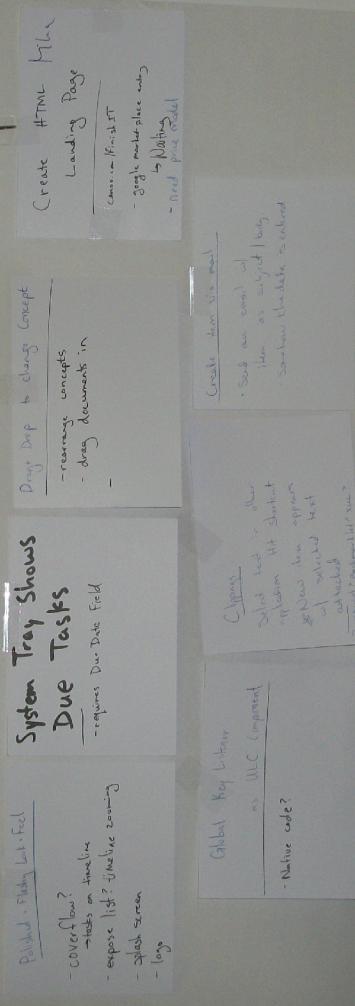
"We have a GAE App"



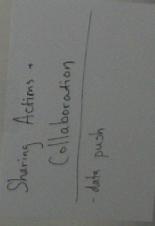
FINISH
►canoo►

"We have an App we can Demo / Showcase"

- things no one else can do



We have the Best in the World



Ready to Start

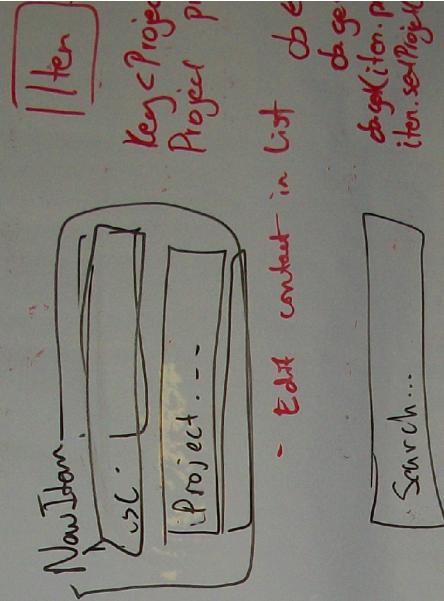
Group Items
on Headers
USE
ONE-TO-ONE
Question

In Development

Change Sheets
of Specific
Infrastructure
DUE DATE

In Polishing

- Clear Labels Bring → should be



**MAKE LIST
OF (FEATURE)
REQUESTS TO
ULL - TEAM**

Item expand
on selection

Build
Integration

Take care
of differences
between
old and new
version
Stay the
same after
switch

Changing browser
lets "kill" the
whole application

Widget

In Testing

Flow of Value

Req.

Due Date

Fix

- Assign Item to category
- Create Contact

→ Released: ← Awesome!!

- Delete Contact
- Edit Contact
- Bind Contact Groups

List x = new ArrayList<String>;
add ("foo"); add ("bar");

33;

Spalte

Contacts
Integration

Display
Login
Screen

Client
Log In Log Out

isDone
Checklist
+ Filter

Project
Support
+ Management
Support
+ Management
Queries

Display
Inbox

Build
Nodes
- Old static links
- New dynamic
links
- New static
links

Widget

LLC Webkit Links
- Causes error
- Solution: exception
between static and dynamic

Create
Task

Add
sample
data
! lab account!

USE
Lookup
Field

Contact
Management + Print
Report

Management
of DB & Core
Queries

Contract
Management + Print
Report

Management
Support

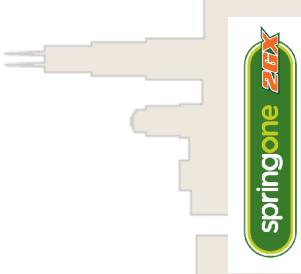
Support
+ Management
Queries

Support
+ Management
Support

Queuing Theory

Delivery becomes slower as...

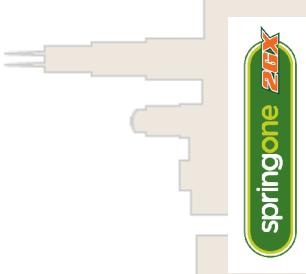
- variability increases
- batch size increases
- utilization increases



Little's Law and Throughput

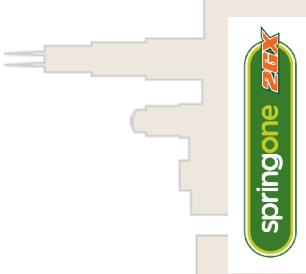
$$\text{Inventory} = \text{Arrival Rate} \times \text{Flow Time}$$

- 5 People in Ramp Jam (Inventory)
- 1 Minutes Per Person (Flow Time)
- Wait is 5 minutes (Arrival Rate)



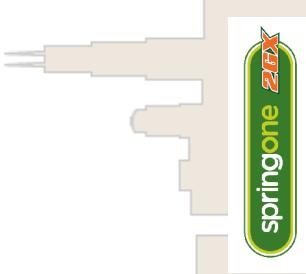
Little's Law and Throughput

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



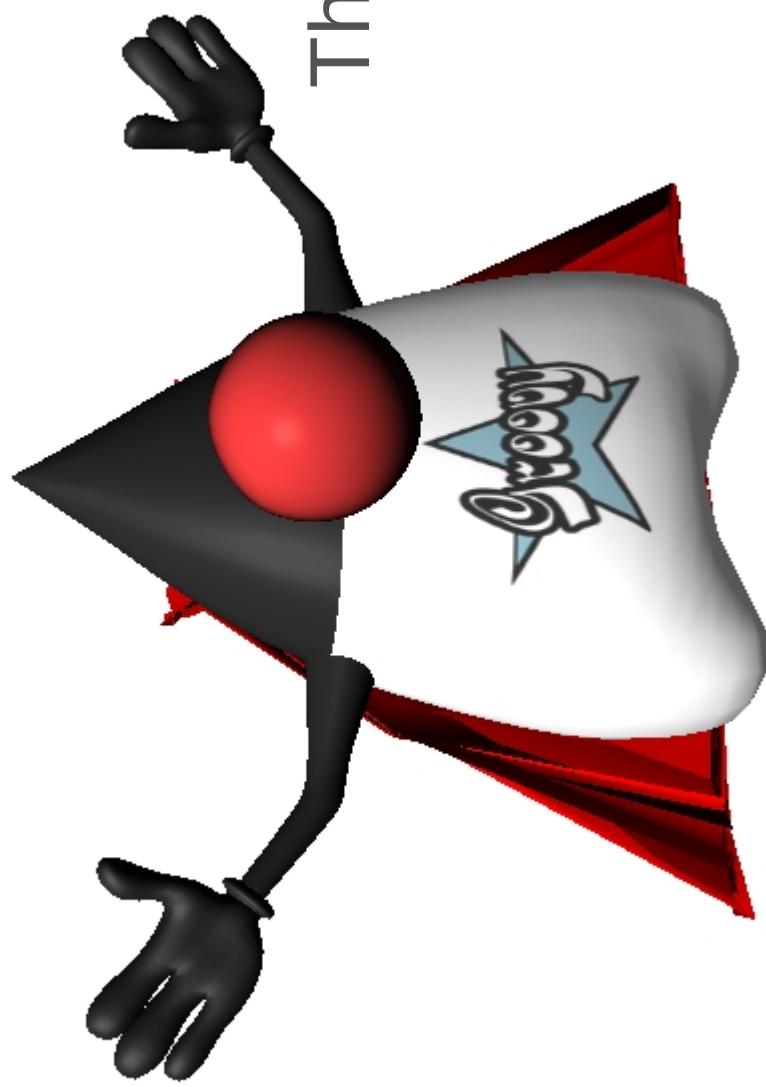
Little's Law and Throughput

- work in small batches
- create slack time
- reduce variability

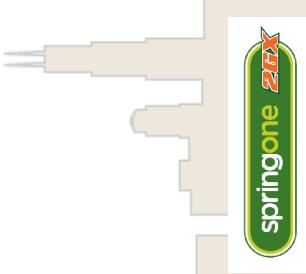




The Small Batch Language



Are your automated tests
helping you go faster?



Platform Services

תְּבִרְכָּה

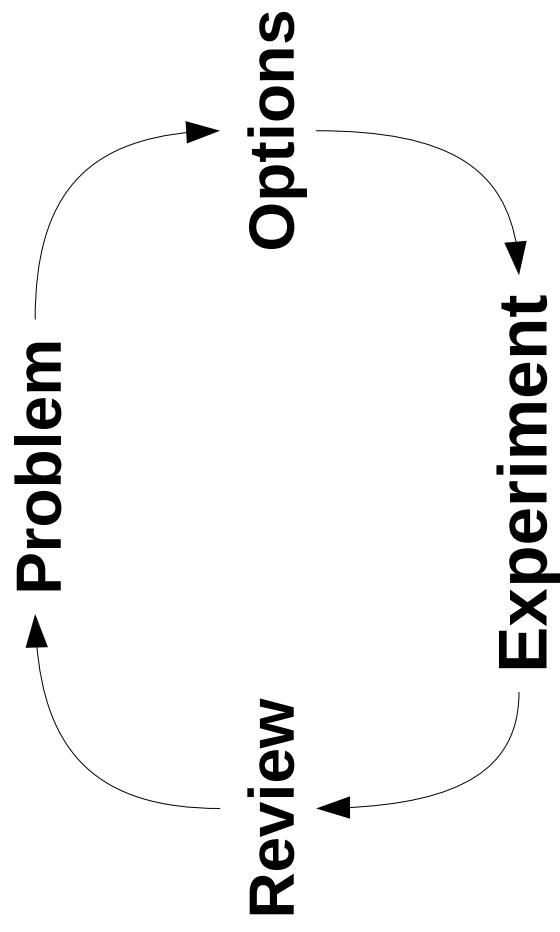
Was sind Goldkarten?	Wie werden Goldkarten benannt?
→ Zeitfenster	→ Jeder SW ist ein Goldkarte
→ Gold Cards	→ Eine Goldkarte
→ Der SW ist	→ Eine Goldkarte
	→ Die Result

FRANK		4.		5.			
Brotfitter		Sprint		Sprint		Sprint	
Profile	JavaSoft 4 Remote	14.07.10 - 27.07.10	28.07.10 - 10.08.10	14.07.10 - 27.07.10	28.07.10 - 10.08.10	14.07.10 - 27.07.10	28.07.10 - 10.08.10
Brotfitter	X	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10
6.		7.		10.		11.	
Sprint		Sprint		Sprint		Sprint	
JavaSoft 4 Remote	Profile	11.08.10 - 24.08.10	25.08.10 - 07.09.10	06.10.10 - 19.10.10	20.10.10 - 02.11.10	03.11.10 - 16.11.10	17.11.10 - 30.11.10
Brotfitter	X	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10
15.		16.		19.		20.	
Sprint		Sprint		Sprint		Sprint	
JavaSoft 4 Remote	Profile	11.08.10 - 24.08.10	25.08.10 - 07.09.10	06.10.10 - 19.10.10	20.10.10 - 02.11.10	03.11.10 - 16.11.10	17.11.10 - 30.11.10
Brotfitter	X	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10
10.		11.		12.		13.	
Sprint		Sprint		Sprint		Sprint	
JavaSoft 4 Remote	Profile	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10
Brotfitter	X	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10
14.		15.		16.		17.	
Sprint		Sprint		Sprint		Sprint	
JavaSoft 4 Remote	Profile	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10
Brotfitter	X	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10
17.		18.		19.		20.	
Sprint		Sprint		Sprint		Sprint	
JavaSoft 4 Remote	Profile	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10
Brotfitter	X	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10
21.		22.		23.		24.	
Sprint		Sprint		Sprint		Sprint	
JavaSoft 4 Remote	Profile	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10
Brotfitter	X	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10

Thomas		Max		Sprint		Sprint		Sprint		Sprint		Sprint		Sprint														
1.	Sprint	3.	Sprint	5.	Sprint	6.	Sprint	7.	Sprint	9.	Sprint	10.	Sprint	11.	Sprint	12.	Sprint	13.	Sprint	14.	Sprint							
02.06.10 - 15.06.10	 Max <small>Max Sprint</small>	30.06.10 - 13.07.10	 Sprint <small>Sprint</small>	28.07.10 - 10.08.10	 Sprint <small>Sprint</small>	11.08.10 - 24.08.10	25.08.10 - 07.09.10	22.09.10 - 05.10.10	22.09.10 - 05.10.10	17.11.10 - 30.11.10	17.11.10 - 30.11.10	01.12.10 - 14.12.10	01.12.10 - 14.12.10	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	Sprint	Sprint	Sprint	Sprint	23.	Sprint	Sprint	Sprint	Sprint
06.10.10 - 19.10.10	20.10.10 - 02.11.10	03.11.10 - 16.11.10	11.	Sprint	16.	Sprint	17.	Sprint	18.	Sprint	20.	Sprint	21.	Sprint	22.	Sprint	23.	Sprint	Sprint	Sprint	Sprint							

Alex	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.
	Sprint										
	14.07.10 - 27.07.10	28.07.10 - 10.08.10	11.08.10 - 24.08.10	25.08.10 - 07.09.10	26.08.10 - 09.09.10	22.09.10 - 05.10.10	06.10.10 - 19.10.10	07.10.10 - 20.10.10	03.11.10 - 16.11.10	17.11.10 - 30.11.10	01.12.10 - 14.12.10
Alba A	JUNIOR	Soap II	Soap III	Soap IV	Soap V	Soap VI	Soap VII	Soap VIII	Soap IX	Soap X	Soap XI

Recipes for Going Faster



From “Agile Coaching”
(a very good book)

Recipe for Disaster:

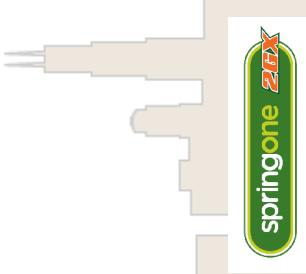
increase productivity in a non-bottleneck area



Next Steps: Profit and Loss Sheet

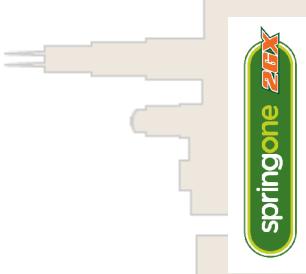
What has the greatest cost?

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



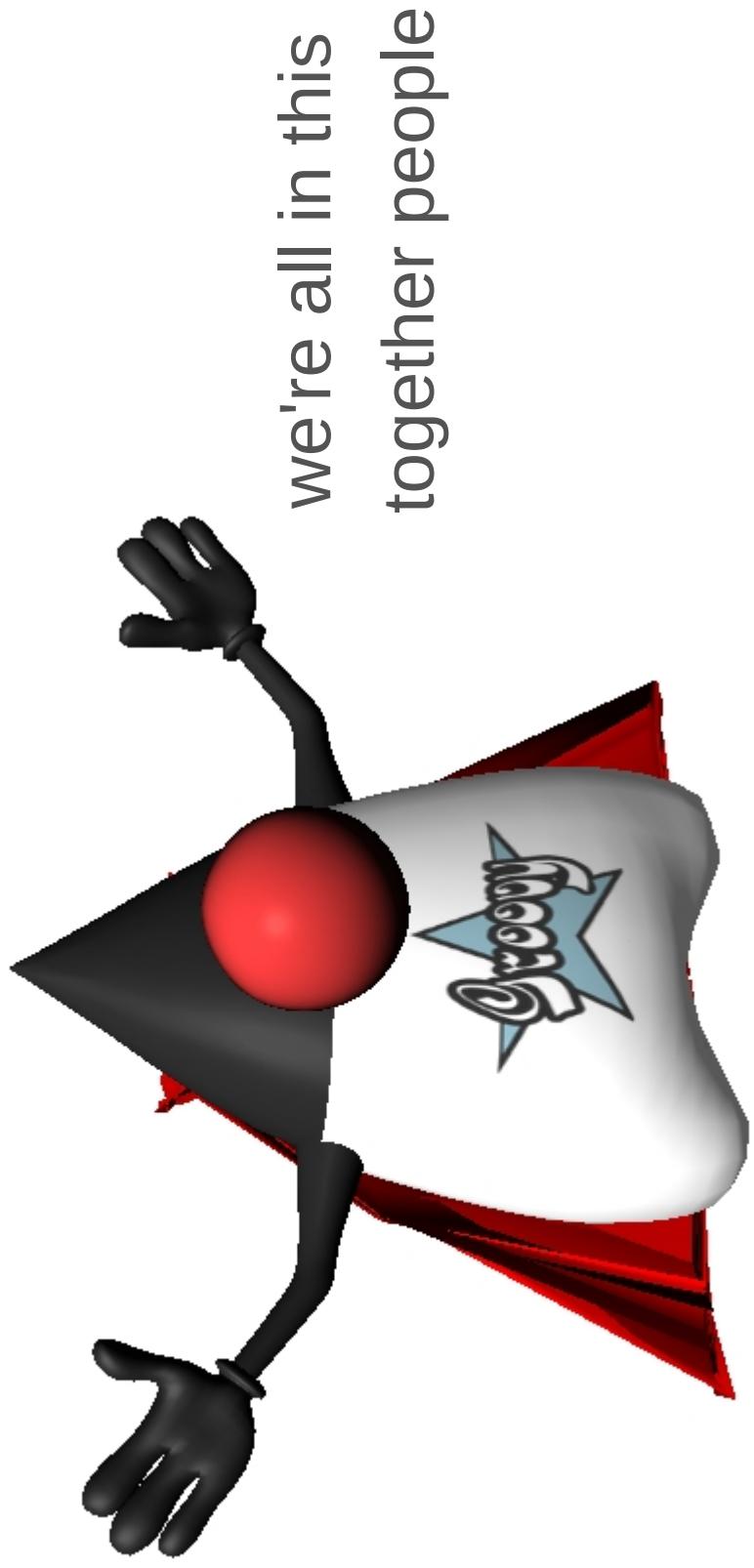
Lean Principles

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

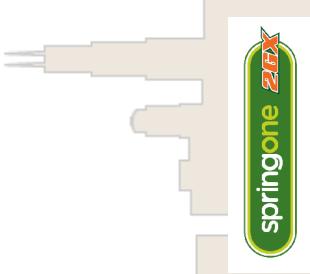


... some anti patterns



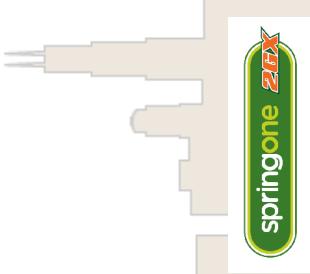


we're all in this
together people



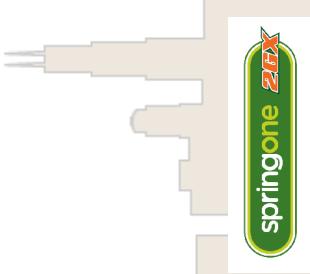
“Changing culture is hard, but for many organizations,
it’s the critical first step towards better software”

Nathaniel Schutta



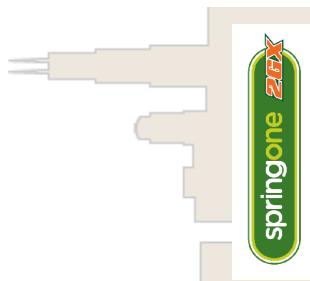
Agile Adoption Stage 0

- 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

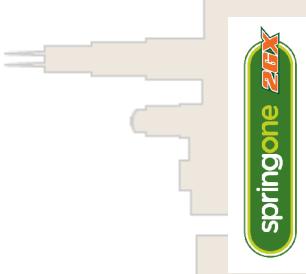


Demo

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

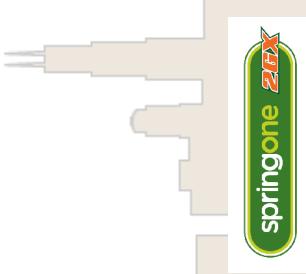


“You're free to
use Groovy for
personal scripting”



Lean Principles

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



Documentation Breeds in Captivity

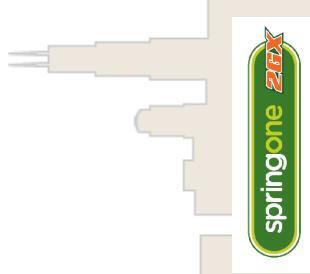
Which is the canonical source?

Requirements

User Stories

Functional Tests

Test Plans



Demo

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



automate, automate, automate



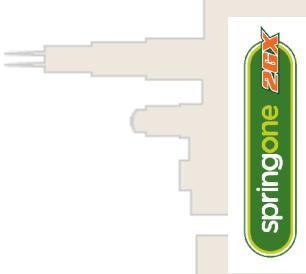
'automate' * 3
better?



Gradle

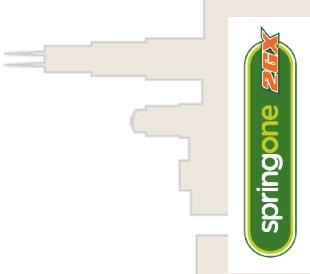
a better way to build

A platform for automation
(and covered better elsewhere)



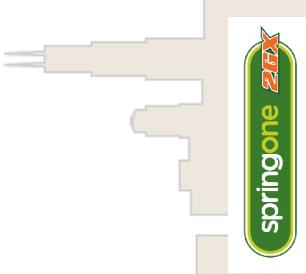
Lean Principles

- 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

