Damon Poole

- Chief Agilist, Eliassen Group’s Agile Practice
- Creator of multiple Jolt-award winning products
- 20 years of process improvement ranging from small co-located teams to 80 team global development shops.
- Past President of Agile New England
- Author of “DIY Agile Kickstart”
- Founder and past CTO and CEO of AccuRev
WE'RE GOING TO TRY SOMETHING CALLED AGILE PROGRAMMING.
Agenda

- Introduction
- Overview of Agile
- Being an Agile Programmer
- User Stories
- Backlog
- Story Splitting
- Visualize your work
- Limited Work in Progress
- Unit testing
- Refactoring
- TDD
- Continuous Integration
We are uncovering better ways of developing software by doing it and helping others do it.

**Values**

“Working software over comprehensive documentation”

**Principles**

“Working software is the primary measure of progress.”
We are uncovering better ways of developing software by doing it and helping others do it.

**Values**

- "Individuals and interactions over process and tools"
- "Working software over comprehensive documentation"
- "Customer collaboration over contract negotiation"
- "Responding to change over following a plan"

**Principles**

- "Build projects around motivated individuals, give them the environment and support they need, and trust them to get the job done,"
- "Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely."
- "The best architectures, requirements, and designs emerge from self-organizing teams"
- "Continuous attention to technical excellence and good design enhances agility."
- "Working software is the primary measure of progress."
- "Simplicity -- the art of maximizing the amount of work not done -- is essential."
- "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software."
- "Working software is the primary measure of progress."
- "Simplicity -- the art of maximizing the amount of work not done -- is essential."
- "At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly."
- "Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage."

Scrum?

Kanban?

XP?
Many Practices are Subject of a Whole Book!
What’s in a Feature?

Show loan status.
Each Feature is Comprised of Many Aspects

```c
#define a b

/*
 * This is great code.
 */

bool doStuff() {
    index++;
}
```
Every Project Contains Features of Many Sizes
Traditional Development, More or Less

- Risk reduced
- Actual cost known
- Feedback available
- Problems known
- Able to change course
- Able to receive value
Agile Development

- Risk reduced
- Actual cost known
- Feedback available
- Problems known
- Able to change course
- Able to receive value

Higher Quality
Options
Visibility

Twitter

Higher ROI Faster
Operational Efficiency vs Business Value Efficiency

Done
Agenda

- Introduction
- Overview of Agile
- Being an Agile Programmer
- User Stories
- Backlog
- Story Splitting
- Visualize your work
- Limited Work in Progress
- Unit testing
- Refactoring
- TDD
- Continuous Integration
<table>
<thead>
<tr>
<th>Column A</th>
<th>Column B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not able to take all vacation days</td>
<td>• Not able to take all vacation days</td>
</tr>
<tr>
<td>Not clear how work provides value</td>
<td>• Not clear how work provides value</td>
</tr>
<tr>
<td>Working on boring projects</td>
<td>• Working on boring projects</td>
</tr>
<tr>
<td>Working on old technology</td>
<td>• Working on old technology</td>
</tr>
<tr>
<td>Declining working conditions</td>
<td>• Declining working conditions</td>
</tr>
<tr>
<td>Company losing competitiveness</td>
<td>• Company losing competitiveness</td>
</tr>
<tr>
<td>Company value declining</td>
<td>• Company value declining</td>
</tr>
<tr>
<td>Not getting bonuses</td>
<td>• Not getting bonuses</td>
</tr>
<tr>
<td>Benefit reduction</td>
<td>• Benefit reduction</td>
</tr>
<tr>
<td>Salary reduction</td>
<td>• Salary reduction</td>
</tr>
<tr>
<td>Working long hours</td>
<td>• Working long hours</td>
</tr>
<tr>
<td>Project shelved</td>
<td>• Project shelved</td>
</tr>
<tr>
<td>Project cancelled</td>
<td>• Project cancelled</td>
</tr>
<tr>
<td>Layoffs/RIFs</td>
<td>• Layoffs/RIFs</td>
</tr>
<tr>
<td>Out of business</td>
<td>• Out of business</td>
</tr>
</tbody>
</table>

© 2012 Valtivity and others. All Rights Reserved - 17-
Market/Customers

Product Mgmt

Strategy

Todo

Results

Customer Value

Tangible Value

Revenue

Bank Account

Cancelled

Shelved

Unused

old

new
Agenda

- Introduction
- Overview of Agile
- Being an Agile Programmer
- User Stories
- Backlog
- Story Splitting
- Visualize your work
- Limited Work in Progress
- Unit testing
- Refactoring
- TDD
- Continuous Integration
- **<user role>** wants to **<achieve some goal>**

  - **Traveller wants to enter a booking**
  - **Airline wants data on people booking hotels with flights**
  - **Traveller wants to edit a booking**
  - **Admin wants a report of site-wide activity**
  - **“Customer collaboration over contract negotiation”**
  - **“Business people and developers must work together daily throughout the project.”**
User Story

Traveller wants to edit a booking

8

Who is working on it

Customer

Headline

Estimate in story points

Goal

Tom
User Story

Traveller wants to edit a booking

- Customer focused
- Customer, customer proxy, scrum master, developer, tester, and documenter can all understand them
- Separates the “what” from the “how”
Who is “The Customer?”

- Market/Customers
- Business Unit
- Software Development
Who is “The Customer?”

Market/Customers

Your offering
Bill Wake’s “INVEST” Guidelines

- Independent
  - Aside from core functionality, doesn’t depend on other stories

- Negotiable.
  - A story is a conversation starter, not the end result

- Valuable to the user, can be used directly
  - “Implement Avatar api” has no value in and of itself

- Estimable
  - No research required, well understood

- Small

- Testable
Agenda

- Introduction
- Overview of Agile
- Being an Agile Programmer
- User Stories
- Backlog
- Story Splitting
- Visualize your work
- Limited Work in Progress
- Unit testing
- Refactoring
- TDD
- Continuous Integration
Who wants to make some money?

Cost

Value
The Time Value of Money

Learning also has value
## Managing Priorities

<table>
<thead>
<tr>
<th>Priority</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Committed to release</td>
</tr>
<tr>
<td>1</td>
<td>As soon as possible</td>
</tr>
<tr>
<td>2</td>
<td>Important</td>
</tr>
<tr>
<td>3</td>
<td>Nice to have</td>
</tr>
<tr>
<td>4</td>
<td>Not important</td>
</tr>
</tbody>
</table>
# Managing Priorities

<table>
<thead>
<tr>
<th>Priority</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Maybe</td>
</tr>
</tbody>
</table>
## Managing Priorities

<table>
<thead>
<tr>
<th>Priority</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Maybe</td>
</tr>
<tr>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>No</td>
</tr>
</tbody>
</table>
Backlog

Most Value

- Traveller wants to enter a booking (Bob)
- Traveller wants to see their upcoming trips (Tom)
- Traveller wants to edit a booking (Tom)
- Traveller wants to delete a booking (Sue)
- Traveller wants to copy a booking (Bob)
- Admin wants a report of site-wide activity (Bob)
- Traveller wants to move a booking (Bob)
- Traveller wants to link to cancel a booking (Bob)
- Traveller wants to link to on-line check-in (Bob)

“Responding to change over following a plan”

“Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.”

“Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.”

“Simplicity -- the art of maximizing the amount of work not done – is essential.”

Least Value
Negotiation Tool

Traveller wants to e-mail an airline booking

Traveller wants to see their upcoming trips

Traveller wants to edit a booking

Traveller wants to delete a booking

Traveller wants to copy a booking

Admin wants a report of site-wide activity

Traveller wants to move a booking

Traveller wants to link to cancel a booking

Traveller wants to link to online check-in

Release

© 2012 Valtivity and others. All Rights Reserved - 33-
Agenda

- Introduction
- Overview of Agile
- Being an Agile Programmer
- User Stories
- Backlog
- Story Splitting
- Visualize your work
- Limited Work in Progress
- Unit testing
- Refactoring
- TDD
- Continuous Integration
Splitting out the Gold

User wants…

“Yes! I need that and can use it.”
Operational Efficiency vs. Business Value Efficiency

Value vs. Cost/Time
Splitting out the Gold

User wants… 3

“Yes! I need that and can use it.”

User wants… 5

“Yes! I need that and can use it.”
Done
Operational Efficiency vs. Business Value Efficiency
Story Splitting Techniques

- At “and” or “comma”
- By acceptance Test
- By user
- By grafting new technology onto old
- By workflow
- By level of value / constraining effort
- By numerical reduction
- Into Create/Read/Update/Delete
- By “going sideways”
- By use case
Splitting by Acceptance Test

Acceptance Tests
1. Handles a hotel booking
2. Handles an airline booking
3. Handles a car booking
Breaking Down by Layer

- Person wants 13 UI for sending a greeting card
- Person wants 13 middleware for sending a greeting card
- Person wants 13 back end for sending a greeting card

© 2012 Valtivity and others. All Rights Reserved - 43-
Splitting By Workflow

Person wants to select a card to send 5
Person wants to customize the card 5
Person wants to check a proof of the card 5
Person wants to select delivery options 5
Person wants to pay for the card 5
Person wants error checking 5

Velocity: 20
Person wants to select a card to send

GUI for card selection

Xmit card type to server

Artwork

Textbox for card message

Xmit custom information

Person wants to customize the card

GUI for card customization

Textbox for recipient name

Display final card

Generate final card

Person wants to check a proof of the card

GUI for proofing

Textbox for address

Xmit delivery options

GUI for delivery options

Validate payment information

Create card fulfillment data

Person wants to select delivery options

GUI for delivery options

Xmit delivery options

Place card order with partner

Person wants to pay for the card

GUI for payment options

© 2012 Valtivity and others. All Rights Reserved - 45-
Person wants to send a happy birthday card

Set card type as HAPPY_BDAY
GUI to collect Happy Birthday info
GUI for card selection
GUI for card customization

Xmit card type to server
Textbox for recipient name
Textbox for address

Textbox for card message
Xmit custom information

Display final card
Generate final card

GUI for proofing
GUI for delivery options
Xmit delivery options

GUI for payment options
Validate payment information
Create card fulfillment data
Place card order with partner

© 2012 Valtivity and others. All Rights Reserved
Send a “Happy Birthday” message

To: Grandma Betty

Address: 3 New England Drive
         Boston, |

Ok  Cancel

Person wants to send a happy birthday card
Person wants to send a happy birthday card

- Person wants to select a card to send
- Person wants to customize the card
- Person wants to select delivery options
- Person wants to check a proof of the card
- Person wants to pay for the card
<table>
<thead>
<tr>
<th>Person wants to select a card to send</th>
<th>Person wants to customize the card</th>
<th>Person wants to check a proof of the card</th>
<th>Person wants to select delivery options</th>
<th>Person wants to pay for the card</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI for card selection</td>
<td>GUI for card customization</td>
<td>GUI for proofing</td>
<td>GUI for delivery options</td>
<td>GUI for payment options</td>
</tr>
<tr>
<td>Xmit card type to server</td>
<td>Textbox for recipient name</td>
<td>Display final card</td>
<td>Xmit delivery options</td>
<td>Validate payment information</td>
</tr>
<tr>
<td>Xmit card example to client</td>
<td>Textbox for address</td>
<td>Generate final card</td>
<td></td>
<td>Create card fulfillment data</td>
</tr>
<tr>
<td>Artwork</td>
<td>Textbox for card message</td>
<td></td>
<td></td>
<td>Place card order with partner</td>
</tr>
<tr>
<td>Xmit custom information</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© 2012 Valtivity and others. All Rights Reserved - 49-
Person wants to select a card to send

GUI for card selection

Xmit card type to server

Artwork

Person wants to customize the card

GUI for card customization

Textbox for recipient name

Xmit card example to client

Textbox for address

Textbox for card message

Xmit custom information

Person wants to check a proof of the card

GUI for proofing

Display final card

Generate final card

Project wants to select delivery options

GUI for delivery options

Xmit delivery options

GUI for payment options

Validate payment information

Create card fulfillment data

Place card order with partner
Person wants to select a card to send

GUI for card selection

Xmit card type to server

Artwork

Xmit card example to client

Person wants to customize the card

GUI for card customization

Textbox for recipient name

Textbox for card message

Xmit custom information

Person wants to check a proof of the card

GUI for proofing

Display final card

Generate final card

Xmit delivery options

Place card order with partner

Person wants to select delivery options

GUI for delivery options

Validate payment information

Create card fulfillment data

Person wants to pay for the card

GUI for payment options
Agenda

- Introduction
- Overview of Agile
- Being an Agile Programmer
- User Stories
- Backlog
- Story Splitting
- Visualize your work
- Limited Work in Progress
- Unit testing
- Refactoring
- TDD
- Continuous Integration
Traveller wants to see their upcoming trips.

Admin wants a report of site-wide activity.

Bob wants to enter a booking.

Bob wants to register with the system.

Sue wants to copy a booking.

Sue wants to e-mail a hotel booking.

Sue wants to e-mail a car booking.

Sue wants to e-mail an airline booking.

Hotel owner wants to check usage data.

Airline wants to check usage data.

Car rental agency wants to check usage data.

Seller wants to remove an ad.

Seller wants to show an ad.

Traveller wants to link to on-line check-in.

Traveller wants to link to cancel a booking.

© 2012 Valtivity and others. All Rights Reserved - 53-
backlog → iter → coding → testing → accept → done

- Seller wants to remove an ad
- Seller wants to change an ad
- Traveller wants
- Traveller wants
- Traveller wants
- Traveller wants
- Hotel owner wants
to check
- Airline wants
to check
- Car rental agency wants
to check

- Traveller wants
- Admin wants a report of site
- Traveller wants
- Traveller wants
- Traveller wants
- Traveller wants
- Traveller wants
- Traveller wants
- Traveller wants
- Traveller wants
- Traveller wants

- Bob
- Sue
- Tom
- Sue
- Bob
- Sue
- Bob
- Sue
- Bob
- Sue

© 2012 Valtivity and others. All Rights Reserved - 54-
Seller wants to remove an ad 5
Seller wants to show an ad 5
Traveller wants 2
Traveller wants 2
Traveller wants 2
Traveller wants 2
Hotel owner wants to check usage data 2
Airline wants to check usage data 2
Car rental agency wants to check usage data 2

Traveller wants 2
Traveller wants 1
Traveller wants 2
Traveller wants 2
Traveller wants 3
Traveller wants to register with the system Sue 3

Traveller wants to see their upcoming Tom trips 2
Admin wants a report of site-wide activity Bob 5

Traveller wants to e-mail a hotel booking 2

© 2012 Valtivity and others. All Rights Reserved - 55-
Seller wants to remove an ad
Seller wants to show an ad
Traveller wants
Traveller wants
Traveller wants
Traveller wants
Hotel owner wants to
Airline wants to check
Car rental agency wants to check usage data

Traveller wants to edit a booking Tom
Traveller wants to see their upcoming Tom trips
Traveller wants to e-mail a hotel booking
Admin wants a report of site-wide activity Bob
Traveller wants to copy a booking Bob

Traveller wants to enter a booking Bob
Traveller wants to register with the system Sue
Traveller wants to delete a booking Sue
Activity vs Achievement
Seller wants to remove an ad
Seller wants to show an ad
Traveller wants
Traveller wants
Traveller wants
Hotel owner wants to
Airline wants to
Car rental agency wants to check usage data

Traveller wants to edit a booking
Traveller wants to see their upcoming trips
Admin wants a report of site-wide activity
Traveller wants to e-mail a hotel booking
Traveller wants to e-mail an airline booking
Traveller wants to copy a booking

Traveller wants to enter a booking
Traveller wants to register with the system
Traveller wants to delete a booking
Traveller wants to link to on-line check-in
Traveller wants to link to cancel a booking

Hotel owner wants to check usage data
Airline wants to check usage data
Car rental agency wants to check usage data
Seller wants to remove an ad
Seller wants to show an ad
Traveller wants
Traveller wants
Traveller wants
Traveller wants
Hotel owner wants to
Airline wants to check
Car rental agency wants to check usage data

Traveller wants to edit a booking
Traveller wants to e-mail a hotel booking
Traveller wants to copy a booking

Admin wants a report of site-wide activity

Traveller wants to enter a booking
Traveller wants to delete a booking
Traveller wants to see their upcoming trips

Traveller wants to register with the system
Seller wants to remove an ad
Seller wants to show an ad
Traveller wants
Traveller wants
Traveller wants
Traveller wants
Hotel owner wants to
Airline wants to
Car rental agency wants to check usage data

Traveller wants to edit a booking Tom
Traveller wants to e-mail a hotel booking
Traveller wants to enter a booking Bob
Admin wants a report of site-wide activity Bob
Traveller wants to copy a booking Bob

Traveller wants to register with the system Sue
Traveller wants to delete a booking Sue

Traveller wants to see their upcoming trips Tom
seller wants to remove an ad
seller wants to show an ad
traveller wants
traveller wants
traveller wants
hotel owner wants
time airline wants
airline wants
car rental agency wants to check usage data
traveller wants to see their upcoming trips
tom admin wants a report of site-wide activity
bob traveller wants to email a hotel booking
bob traveller wants to edit a booking
sue traveller wants to delete a booking
sue traveller wants to copy a booking
bob traveller wants to enter a booking
bob traveller wants to register with the system
sue traveller wants to link to on-line check-in
tom traveller wants to link to cancel a booking
airline wants to check usage data
car rental agency wants to check usage data
Agenda

- Introduction
- Overview of Agile
- Being an Agile Programmer
- User Stories
- Backlog
- Story Splitting
- Visualize your work
- Limited Work in Progress
- Unit testing
- Refactoring
- TDD
- Continuous Integration
Bugs are NOT OK
Bugs Displace Value

Release A

Release B
Bug fixes
Deferred from B

Release C
Bug fixes
Deferred from C

Release D
Typical Test Cycle

1. Thinking
2. Exploratory testing
3. Writing of test cases
4. Writing of manual test scripts
5. Automation
6. Run tests
7. Update regression suite
8. Throw away
Moving away from concept of “regression” testing

1. Thinking
2. Writing of test cases
3. Exploratory testing
4. Writing of manual test scripts
5. Automation
6. Update test suite
7. Run tests continuously
- Open/Closed pattern from OO programming

Refactor
Unitized Testing

- Time spent writing unit tests and refactoring replaces time spent debugging and manual testing
Holistic testing

9 UI test cases

Front end

Middle layer

Back end, infrastructure
Holistic Testing

9 total test cases

Front end (QTP+Unit)

Middle layer (SoapUI+Unit)

Back end, infrastructure (Unit+others)
Unit Test Opportunities

- UI code
- Services
- Middle tier code
- Server code
- Database: triggers, stored procedures
- Deployment activities
Agenda

- Introduction
- Overview of Agile
- Being an Agile Programmer
- User Stories
- Backlog
- Story Splitting
- Visualize your work
- Limited Work in Progress
- Unit testing
- Refactoring
- TDD
- Continuous Integration
Test Driven Development

- Programmers write unit tests prior to coding
- If you can’t write a test, you aren’t ready to code
- Writing tests first will change how you write code
- Writing tests after you have written the code is “too late”
- Writing test cases (not test scripts) prior to writing should also be considered
Agenda

- Introduction
- Overview of Agile
- Being an Agile Programmer
- User Stories
- Backlog
- Story Splitting
- Visualize your work
- Limited Work in Progress
- Unit testing
- Refactoring
- TDD
- Continuous Integration
All Tests Run All the Time

- Reduces the risk of making a change
- Reduced risk enables moving quicker
- Problems are found sooner
Continuous Integration

Version Control Check-in → Build Artifacts → Unit testing, static code analysis, etc

Report and notify ← UI, service, db, etc testing ← Deploy

betterCode = unitTesting + refactoring + CI;