Alline Watkins

some enterprises worked:

- Nike
- Citibank
- UCSanDiego
- TAM Brazilian Airlines
- Tata
- Globo.com
- Brazilian Govern

some startups envolved:

- LiveSource
- Cloud Girlfriend
- Iobio
- Fashion Metric
- Lean Startup Machine
- Lifeproof

some presentations at:

- Scrum Master
- Scrum Product Owner certified
80% of the software developed is being wasted.
ENTERPRISES

- Experience
- Large Scale view
- Stability
- Customers ≠ Users

STARTUPS

- Engaged Employees
- Focus on cost reduction
- Adaptability
- Customers = Users

Learn
Lean Startup Topics

- Customer Development
- Uncertainty
- Minimum Viable Product
- Build-Measure-Learn Loop
- Product-Market Fit
Customer Development Tools

Who: Make a Persona

“Mary”

Behaviors
- Has a housecleaner
- Buys take-away 3 nights/week
- Frequently feels overwhelmed when she “forgets” something

Demographics
- Working mom
- 34 years old
- Lives in Reading, works in London
- Married, 2 kids
- Household 12 salary

Needs & Goals
- Help! Running errands, managing kids, keeping things running
- Time for her girlfriends
- To feel like she “has it sorted”
- “To clone herself”

Customer Development Framework

Customer Problem Matrix

<table>
<thead>
<tr>
<th>Customer Problem 1</th>
<th>Customer Problem 2</th>
<th>Customer Problem 3</th>
<th>Your target Problem</th>
<th>“Must Have” Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Segment 1</td>
<td>Customer Name - 4</td>
<td>Customer Name - 4</td>
<td>Customer Name - 2</td>
<td>Customer Name - 4</td>
</tr>
<tr>
<td></td>
<td>Customer Name - 4</td>
<td>Customer Name - 5</td>
<td>Customer Name - 3</td>
<td>Customer Name - 4</td>
</tr>
<tr>
<td></td>
<td>Customer Name - 4</td>
<td>Customer Name - 3</td>
<td>Customer Name - 4</td>
<td>Customer Name - 4</td>
</tr>
<tr>
<td>Customer Segment 2</td>
<td>Customer Name - 5</td>
<td>Customer Name - 4</td>
<td>Customer Name - 4</td>
<td>Customer Name - 4</td>
</tr>
<tr>
<td></td>
<td>Customer Name - 4</td>
<td>Customer Name - 4</td>
<td>Customer Name - 4</td>
<td>Customer Name - 4</td>
</tr>
<tr>
<td></td>
<td>Customer Name - 3</td>
<td>Customer Name - 3</td>
<td>Customer Name - 3</td>
<td>Customer Name - 3</td>
</tr>
<tr>
<td>Customer Segment 3</td>
<td>Customer Name - 4</td>
<td>Customer Name - 3</td>
<td>Customer Name - 3</td>
<td>Customer Name - 5</td>
</tr>
<tr>
<td></td>
<td>Customer Name - 4</td>
<td>Customer Name - 4</td>
<td>Customer Name - 5</td>
<td>Customer Name - 5</td>
</tr>
<tr>
<td></td>
<td>Customer Name - 3</td>
<td>Customer Name - 3</td>
<td>Customer Name - 5</td>
<td>Customer Name - 5</td>
</tr>
</tbody>
</table>

LEAN UX

PROBLEM MATRIX

KANBAN BOARD
for Customer Dev

CustDevFramework.com
"A startup is a human institution designed to deliver a new product or service under conditions of extreme uncertainty."

StartupLessonsLearned.com

"Unless some of you have been working in a specific domain in the last 20 years or so, the odds are anything you are thinking about customers and markets are nothing more than a guess."

Steve Blank
Experiments & Minimum Viable Product

Minimum + Viable
Good features to test the users responses

Minimum
Viable

Crappy Products nobody wants to use

The products you want to build

Waste

Products
"For many events, roughly 80% of the effects come from 20% of the causes." (Pareto Principle)
**Build**

**Minimum** amount of tasks (MVP) (just enough to support the next conversation with the end users)

**Measure**

How the end users are using the software. (early-adopters) (in a production environment)

**Learn**

Stay open to the new directions revealed by end users. (avoid getting stuck to your Product Backlog)

**The Build-Measure-Learn Loop**

*Weekly!*
MINDSETS

at Agile Development ...

I want a software

Ok, let's build the product backlog first

Now I want features A and D

Let's drop features B and C then.

at User Driven Development ...

I want a software

Ok, who is going to use it?

According to the Usage Metrics, next features are X and Z
What do you think?
Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan
- Software being used over fast development

That is, while there is value in the items on the right, we value the items on the left more.
HACKING AGILE
“Lean”, is a production practice that considers the expenditure of resources for any goal other than the creation of value for the end customer to be wasteful, and thus a target for elimination.” (From Wikipedia)

Lean = *preserving value with less work.*
Product Backlog, No More!

Plan for the current iteration

No Upfront Planning
 Agile Planning

Product Backlog

Iteration 1
Iteration 2
Iteration 3

Upfront Planning

Guesses

MVP

Key Features

LOOP 1
LOOP 2
LOOP 3

User Driven Development

EXTREME
"Teams should adjust what they are doing based on what team members learn directly from their efforts."
the product owner role is a stupid idea
how whole teams work together to create successful products

Jeff Patton
jeff@comakewith.us
twitter: @jeffpatton
<table>
<thead>
<tr>
<th>AGILE Meetings</th>
<th>USER DRIVEN DEV Meetings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sprint Planning</strong> (4 hours / 2 weeks)</td>
<td>Just don't do it.</td>
</tr>
<tr>
<td><strong>Tasking meeting</strong> (4 hours / 2 weeks)</td>
<td>Tasking while developing. On the fly.</td>
</tr>
<tr>
<td><strong>Estimation meeting</strong> (1 or 2 hours / 2 weeks)</td>
<td>Estimate while developing. On the fly.</td>
</tr>
<tr>
<td><strong>Stand-up meetings</strong> (2 to 4 hours / 2 weeks)</td>
<td><strong>Improvements meeting</strong> (3 hours / week)</td>
</tr>
<tr>
<td><strong>Sprint Demo</strong> (4 hours / 2 weeks)</td>
<td><strong>Usability tests</strong> (3 hours / week)</td>
</tr>
<tr>
<td><strong>Retrospective</strong> (3 hours / 2 weeks)</td>
<td><strong>Improvements meeting</strong></td>
</tr>
</tbody>
</table>

20 to 23 hours / 2 weeks sprint | 6 hours / week
The **UDD** Process

- Inputs from Executives, Team, Stakeholders, Customers, Users
- Improvements Meeting
- Development + Users + Stakeholders
- Development
- User Experience
- Metrics Analysis

2-3 Days Loop
Ideation

Inputs from Executives, Team, Stakeholders, Customers, Users

Product Owner

Vision

MVP

Lean UX

UX Vision
UX Principals
Personas
Scenarios
User Stories
Small Improvements Meeting

"Startup" Team

AUTONOMY
MASTERY
PURPOSE

No Hierarchy

Developers + Users + Stakeholders

It's OK to Experiment

改
KAI
善
ZEN

Change is Good
Task Breakout

Just let Developers do it in real-time.

It’s not just about time... It’s also about relevance.
Development

Full-time status exposed
User Experience

Real User + Real Data

Developer

Production Environment

User + Developer

Early Adopters
Metrics

Visits
Clicks
Usage
Returns
Payments
Logins
Satisfaction
Access
Performance

INNOVATION  ACCOUNTING
No backlog

No Hierarchy

No Tasking

No estimation

No Def of Done

Waste Not

Minimum Viable Product

"Startup" Team

Small Improvements

Task Recycling

Usage Metrics
Thank you Alline

After your MVP and 20% 80% talk
I decided to scrap doing ecommerce for my store (for now) and launch with just paypal buy button in place...... and look, large chunks of the software I thought I needed now do not need to be in place for me to start selling

It's not UDD but it was still inspired by your two user group talks this month.
A Real Lean Enterprise

Enterprise software
Innovative
Uncertain
Agile Methodology
Developers + Users
Well Defined Vision
Lean UX
Customers Interviews
Minimum Viable Product
Build-Measure-Learn Loop

http://www.youtube.com/watch?v=szr0ezLyQHY
FEEDBACK, PLEASE!

UserDrivenDev.com
Thank you

alline.oliveira@gmail.com
"You gotta start with the **customer experience** and works backward to the technology. You can’t start with the technology and try to figure out where you are going to sell it."

"Some **mistakes will be made** along the way. That’s good because at least some decisions are being made along the way. And we’ll find the mistakes and we’ll fix them."

---

**Steve Jobs, WWDC 1997**