Waterfall to Agile

UCSB CS 189a ● January 13, 2015

Colin Kelley
CTO & Co-founder ● Invoca, Inc.

colin@invoca.com
My background

Villanova University, ’87: BSEE/CS

Forum Systems ’87-’88: Software Developer

Digital Sound Corp / PulsePoint / Unisys ’90s: Software Architect

CallWave, Inc ’00s: CTO

RingRevenue, Inc ’08 +: CTO & Cofounder

Now Invoca, Inc!
### Size of Teams I’ve Worked With

<table>
<thead>
<tr>
<th>Company</th>
<th>Developers</th>
<th>Total Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gnuplot</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>Forum Systems</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>Digital Sound</td>
<td>25</td>
<td>120</td>
</tr>
<tr>
<td>CallWave</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td>Invoca</td>
<td>20</td>
<td>130</td>
</tr>
</tbody>
</table>
# Team Metrics

<table>
<thead>
<tr>
<th></th>
<th>Auto Test Coverage</th>
<th>Team Integration</th>
<th>Build</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gnuplot</td>
<td>none</td>
<td>-</td>
<td>-</td>
<td>monthly WW</td>
</tr>
<tr>
<td>Forum Systems</td>
<td>none</td>
<td>monthly</td>
<td>monthly</td>
<td>6 months WW</td>
</tr>
<tr>
<td>Digital Sound</td>
<td>none</td>
<td>monthly</td>
<td>monthly</td>
<td>1-2 years WW</td>
</tr>
<tr>
<td></td>
<td>60%</td>
<td>weekly</td>
<td>nightly</td>
<td>yearly WW</td>
</tr>
<tr>
<td>CallWave</td>
<td>80%</td>
<td>continuous</td>
<td>continuous</td>
<td>1-3 months WF</td>
</tr>
<tr>
<td>Invoca</td>
<td>99%</td>
<td>continuous</td>
<td>continuous</td>
<td>1-2 weeks Agile</td>
</tr>
</tbody>
</table>

WW: Weekly, WF: Weekly, Agile
“Wild West”

Just do what needs to be done.

Works with:

Tiny teams
Specialized skills
No time constraints
Self-motivated
“Waterfall”

- Requirements
- Design
- Implementation
- Verification
- Maintenance
“Waterfall”

Phased, top-down approach

Seems to work with:

Medium to large teams
Specialized expertise
Low complexity
Low innovation—done before
Fixed, standardized interfaces
“Waterfall” Falls Short

When you have/need:
Complex, innovative project
Generalists
Fine-tuned interfaces/user experience
Timeboxed, fast time to market
“Waterfall” Falls Short

The 10th edition of the annual CHAOS report from The Standish Group, which researches the reasons for IT project failure in the United States, indicates that project success rates have increased from 15% in 1994 to 34% of all projects in 2004.

Asked for the chief reasons project success rates have improved, Standish Chairman Jim Johnson says,

"The primary reason is the projects have gotten a lot smaller. Doing projects with iterative processing as opposed to the waterfall method, which called for all project requirements to be defined up front, is a major step forward."

source: Jeff Atwood, Coding Horror
“Waterfall” Falls Short

Q: What market research went into the development of the iPad?
A: None. It’s not the consumers’ job to know what they want.
“Waterfall” Falls Short

How do you adjust to change?

How do you estimate schedule? Does the team have any say? Any “buy-in”?

How do you avoid “gold-plating”?

How do you foster teamwork?

Who is responsible for excellence? (Perfectly matching customer needs/wants)
Agile Predecessors

Iterative, Lightweight

Requirements → Design → Implementation → Verification → Maintenance
Agile Influencers

Kanban, Lean Manufacturing

eXtreme Programming

  Test-Driven Development, Continuous Integration,
  Daily stand up, Pair programming, Refactoring

Higher-level, dynamic, meta-programming languages

  DRY = Don’t Repeat Yourself

Software as a Service

Dev-Ops

Getting Things Done by David Allen

Re:work by Jason Fried
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

That is, while there is value in the items on the right, we value the items on the left more.

*source: [www.agilemanifesto.org](http://www.agilemanifesto.org)*
Sashimi Slices
Scrum

Product Backlog → Sprint Backlog → Sprint → Working increment of the software

24 h
30 days
Agile at Invoca

Backlog

Project: Product Backlog
Sprint:
Owner:

Status:
Priority:
Complexity:
Estimate:
Tasks:
Test Results:

Description:
As a network or advertiser I want to split or weight call traffic to different transfer destinations.
As a network or advertiser I want to set up round robin weighting in conjunction with different call center hours operation without having to create a bundled campaign.

Product Owner:
Type:
Total Detail
Estimate:
## Agile at Invoca

### Sprint Stories

<table>
<thead>
<tr>
<th>Title</th>
<th>Priority</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td></td>
<td>12.00</td>
</tr>
<tr>
<td>Beta Report Enhancements</td>
<td></td>
<td>17.00</td>
</tr>
<tr>
<td>Notification Mgt.</td>
<td></td>
<td>6.50</td>
</tr>
<tr>
<td>Display Destination Number in Reports</td>
<td>Low</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Email and Notifications

<table>
<thead>
<tr>
<th>Email Address</th>
<th>Notify</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:cwagner@google.com">cwagner@google.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+Add Additional Email</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Currently receiving these email notifications:

- New advertiser/publisher application
- Possible Fraud alert
- New advertiser call center application
- Budget alerts
- Performance Overview
  - Daily
  - Weekly (every Monday)
  - Monthly (first Monday of the month)

Save
Agile at Invoca Planning
Agile at Invoca

Planning Poker
Agile at Invoca

Planning Poker
tasks with sizes
Agile at Invoca

The best part of planning:

“horse trading”

Then:

COMMIT!
Agile at Invoca

Daily Stand Up
Burndown chart
Agile at Invoca

Demo

As soon as each story is ready, story lead demonstrates it

1. to Development team

2. to Product Owner
Agile at Invoca

System Test (QA)

1. Assign QA owner (who didn’t work on story tasks!)

2. QA owner writes test plan in wiki

3. Story owner reviews plan

4. QA owner runs it
Agile at Invoca

Deploy

Typically at night/weekend

Rotating team of dev-ops

Follow deploy plan written in wiki

Chat over Skype
Sprint 40

Did Well

- Cross training (James)
- Demo ownership (James)
- Took on lots of bugs (Scott)

Do Better

- Add rate limiters to APIs (John)
- Change migration timing to 3 hours. (John)
- **Fix the build with urgency** (Nick)
  - Have a build nanny (Nick)
  - Have more build output. The build data is not available remotely. (Nick)
- Should have demoed Story 10 to Rob.
- When doing post deploy cleanup make sure that we have a clear plan and ownership of the plan. (Scott)
- **QA owner should create post deploy checkout plans.** (Nick)
- **Bring back the usability budget.** (Nick)
- Ensure that the burndown is not overly optimistic, especially at the end. (Colin)
- Add estimates for outstanding bugs. (Colin)
- Leave more time between sprints to finish up odds and ends (Bob, Scott)
Invoca Info Session
Summer + Full Time

Monday, January 26th
5:00 pm

Harold Frank Hall Conference
Room 1132
(next to CSIL)
Re:Work

You need less than you think

Embrace constraints

Throw less at the problem

Launch now

Meetings are toxic

Hire when it hurts

Inspiration is perishable
“Push” systems make sense when supply-limited

“Pull” systems make sense when demand-limited
Lessons from Previous Year’s Capstone

Frontload!

Have weekly MVP milestones. Demo!

Leave more time to prepare presentation & final demo