Agile values, Scrum process: Application to architectural model creation, management and delivery
For every complex problem there is an answer that is clear, simple and wrong.

H.L. Mencken
One Bite at a Time

• Learning Objectives

• Start gathering and utilizing information as a tool to design a better modeling process

• Apply new techniques in estimating and verifying workflow and work product in the architectural model.

• Organize architectural staff in roles more suited to the modeling environment in lieu of the traditional drafting environment

• Mesh these techniques and roles with contractor preconstruction design assist services in Revit
Data Gathering and Analysis

• How many here sign contracts?
• How many of your contracts have basic requirements for model sharing and coordination, or even model creation part of the contract?
• How many contracts require a BIM at turnover?
• How many architects here collect “data beyond that needed to generate business metrics?”
• How many constructors?

WHY NOT?
Data Gathering and Analysis

• How many do LEAN anything?
• How many “sell” LEAN to their clients?
• How many use:
  • Checklists
  • Reviews or retrospectives
  • 5 whys
  • Root cause analysis
  • Pull planning
  • WorkFace Planning / Last Planner
  • S curve charting
Data Gathering and Analysis

Architects collect little to none - beyond business metrics
Data Gathering and Analysis

William Whyte’s the social life of small urban spaces.
Traditional Workflow and Staffing
Traditional Workflow and Staffing

The Architects Handbook of Professional Practice
Traditional Workflow and Staffing

Waterfall

- not Incremental
- Not interphase iterative

Data Gathering and Analysis

• Construction is a COMPLEX SYSTEM we need data to analyze so we can improve our process
Complex Ecosystems

http://www.uvm.edu/renr/nr385se/mod3/complexity.html
Complex Ecosystems

http://www.lnsresearch.com/research-library/research-articles/managing-the-complex-supply-chain-ecosystem
History of LEAN in construction/software

1992 Toyota Production System pub.
1996 Rapid Application Development
1993 "LEAN" Construction coined
1997 Lean Construction Institute
2000 Last Planner System
2001 Toyota Way Published
2001 Agile Manifesto
2004 Teicholz Graph
2006 Managing Quality
2007 WorkFace Planning
2007 IPD Document
2013 Teicholz graph shows no change

NOW
Lean vs Agile

Lean seeks repeatability (reactive adaption)
Agile seeks Reliability (proactive adaption)

Lean was developed as ‘a response to competitive pressures with limited resources.

Agile on the other hand, is a response to complexity brought about by constant change.

Lean is a collection of operational techniques focused on productive use of resources.

Agility is an overall strategy focused on thriving in an unpredictable environment.
Agile

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

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

© 2001, the above authors this declaration may be freely copied in any form, but only in its entirety through this notice.  http://agilemanifesto.org/
Kanban

• literally “card you can see;” signboard or billboard
• It is not a software or management technique it is a method for improving process that came from manufacturing.

• Visualize the workflow (Plan)
• Limit Work in Progress (WIP)
• Measure / Manage flow
• Make policies explicit
• Implement feedback loops
• Improve collaboratively, evolve experimentally

https://en.wikipedia.org/wiki/Kanban_board - jeff.lasovski
Kanban

- Limit your WIP to help accomplish more
- Efficiency: Focus on our value stream. Encourage us to find ways to work with less effort
- Effectiveness: making our options explicit helps make informed decisions.

<table>
<thead>
<tr>
<th>Kanban/scrum board</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backlog/ready</td>
</tr>
</tbody>
</table>

#BIMForumED
Scrum

The Agile: Scrum Framework at a glance

Inputs from Executives, Team, Stakeholders, Customers, Users

Product Owner

The Team

Sprint Backlog

Product Backlog

Sprint Planning Meeting

Task Breakout

Sprint Backlog

1-4 Week Sprint

Sprint end date and team deliverable do not change

Finished Work

Sprint Retrospective

Daily Scrum Meeting

Every 24 Hours

Scrum Master

Burndown/up Charts

www.neonrain.com
Scrum roles
Scrum – User Stories

• Each item, or story, in the product backlog should include the following information:
  • Which users the story will benefit (who it is for)
  • A brief description of the desired functionality (what needs to be built)
  • The reason that this story is valuable (why we should do it)
  • An estimate as to how much work the story requires to implement
  • Acceptance criteria that will help us know when it has been implemented correctly

As a .......... <type of user> I want to .......... <do something> (some business action) because I need to .......... <some value created> (the reason why we did it).
Architectural practice as an Intermediate

Owner (our marketing)

Owner (GC)

Architectural Dev Team

User (the public)

User (Trade Sub)

OWNER (Building)

USERS (Building)

Owner (Owner’s marketing)
Engaging the Construction Industry

<table>
<thead>
<tr>
<th>Construction</th>
<th>Agile</th>
<th>Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall planning</td>
<td>Backlog</td>
<td>Spec TOC &amp; Cartoon</td>
</tr>
<tr>
<td>Weekly Work Plan</td>
<td>Sprint Planning</td>
<td>Sprint Planning</td>
</tr>
<tr>
<td>Work Package</td>
<td>User Story</td>
<td>Specification w/ dwgs</td>
</tr>
<tr>
<td>WorkFace Planning</td>
<td>Story Elaboration</td>
<td>Story Elaboration</td>
</tr>
<tr>
<td>Site Inspection</td>
<td>Customer Showcase</td>
<td>Review of “The Set”</td>
</tr>
<tr>
<td>Weekly Review</td>
<td>Retrospective</td>
<td>Retrospective</td>
</tr>
<tr>
<td>S-Curve</td>
<td>Burn Down Charts</td>
<td>Burn Down Charts</td>
</tr>
<tr>
<td>Look-Ahead Planning</td>
<td>Adaptive Planning</td>
<td>Adaptive Planning</td>
</tr>
</tbody>
</table>
Traditional Roles divided and reorganized

- Principle and Project Manager (essentially unchanged)
- Project Architect (coordinating architecture and built systems)
- Job captain (re-defined)  DM - Database Maintenance
- 3D Modeler, Architectural Drafter
- Seeker/Finder, Intern
- Visualizer
Database Management

• IT management
• Design Applications Manager
• ADBA Application Database Administrator
• DM - Database Maintenance staff (for really large Jobs)
• Content Specialists (Import/Export/Translation)
Application Database Administrator (ADBA)

• ADBA's are responsible for looking after the application tasks pertaining to a specific application. This includes the creation of database objects, snapshots, SQL tuning, etc.
  • Typical ADBA responsibilities:
  • Implement and maintain the database design
  • Create database objects (tables, indexes, etc.)
  • Write database procedures, functions and triggers
  • Assist developers with database activities
  • Tune database queries
  • Monitor application related jobs and data replication activities
Architectural Production Planning and Model Production Planning

Design & QA/QC

Modeling & QA/QC

#BIMForumED
Architecture + DB management + Agile

• On team database management (BIM Lead, BIM Manager)
• Modelers and makers, specializing in object creation and function with architectural training but focused on the long term usefulness of the model. For increased owner value
• Visualizer: not necessarily an architect. Not necessarily a graphic artist. Focused on virtual design and representation.
• ALL team members self organizing and self regulating within a sprint planned by themselves the scrum master and project owner.
IPD – Agile

• What worked, how many of these seem to fit what I’ve talked about
  • Early Involvement of Key Participants
  • Shared Risk and Reward
  • Multi-Party Contract
  • Collaborative Decision Making and Control
  • Liability Waivers Among Key Participants
  • Jointly Developed and Validated Project Goals
  • Mutual Respect and Trust Among Participants
  • Collaborative Innovation
  • Intensified Early Planning

Integrated Project Delivery: Case Studies AIA Calif. Council and AIA National Jonathan Cohen FAIA http://www.aiacc.org/2012/06/05/21042/
Risk reduction

We do fine right now.

With a top down approach (or good guesses), we plan and add a buffer to take care of changes….it works, but it hides the variability, therefore we don’t know what the team can really handle.

Remove the buffers and work out the real WIP.
Risk reduction

“We do this already!”

We have feedback loops, and lots of QA/QC.

Fine. Let’s document your process and see if there are any little things that we can make better, use our documentation to show other teams what they can do.

No process is 100%
Risk reduction

“This new stuff is too risky!”

Where are buffers and contingency for change?
This system is “open” everyone’s view, what if a sprint fails?
If something goes wrong, fix it, resort and “keep moving forward.”
Give the team a safe environment where they are allowed to fail today as long as they learn form it tomorrow. If it was not realistic or if the team bit off more than it could chew, this is feedback to improve planning.
Risk reduction

“It doesn’t work for a business like ours.”

People are used to building large and detailed plans prior to work. Many traditional firms will want a detailed work plan prior to work.

• In my experience most architects don’t like to develop these plans,
• Don’t use them once developed, and even if used
• The rate of change soon makes them unusable.

In design we only have an outline at the best of times, why not take advantage of the opportunity?
Risk Reduction

Test driven development, Pair programming, checklists

Some people will not like or “can’t work” with these tools….they just “slow them down,” they may have a point, examine the situation, the management surrounding them and other team members.

Not every tool is right for every job, for example checklists are more easily accommodated in operational situations.

Perkins + Will is currently planning on building test driven environments for Revit.
Risk reduction

- Using agile management techniques (lean concepts)
- Data gathering
- Analysis to verify and Improve the documentation process
- Rinse and Repeat