Conquering the Architect’s Data Deficit

Marc Chavez FCSI AIA RELi AP CCS CCCA
The way the world works:

For Every complex problem there is an answer
The way the world works:

For Every complex problem there is an answer that is clear, simple, and wrong. - H.L. Mencken
Data gathering and analysis
Data gathering and analysis

- $1230 Billon in United States
  Construction of this 489.3 Billion is Commercial/Institutional

- 50% Gen X Population in A/E firms. Baby Boomers down to 33% from 43% in 2016

- 33 ZB Of collected data in 2018 175 ZB in 2025.

- dead last Amount of data stored per firm >1000 employees in 2009 (231 terabytes) “professional services” = 278
Project Management KPIs Tracked

- Profitability: 94%
- Net revenue: 93%
- Average collection period: 84%
- Multipliers: 80%
- Average billing rate: 66%
- Estimate to complete: 59%
- Effective billing rate: 58%
- Cost variance: 58%
- Estimate at complete: 53%
- Client satisfaction: 46%
- On-time delivery: 42%
- Schedule variance: 33%
- Earned value management: 24%
- Other: 16%

Other KPIs:
- Earned value management
- On-time delivery
- Schedule variance
- Average billing rate
- Average collection period
- Effective billing rate
- Cost variance
- Estimate at complete
- Client satisfaction
- Net revenue
- Multipliers
- Avg billing rate
- Estimate to complete
- Profitability
Data gathering and analysis - KPIs

- Cost variance: 30% are over budget
- Schedule Variance: 36% behind schedule
- Client Satisfaction: 54% of firms are NOT measuring.
- Internal Project Performance Evaluation 51%
- 48% of firms used a clearly defined PM process
- Project leaders with Formal Training: 28% of firms
- Internal resources offered PMs: informal training
- Challenges: Competing priorities, staff, PM expertise
- Initiatives: Clear responsibilities, Best practices, Project Information Mgmt., PM training
- Employee Turnover 13.8% (should be 11%)
Project Management: Next steps

» What’s holding your projects back?

» What do PMs need to more proactively manage projects?

» Are you providing construction feedback to your team?

» Are your processes hindering your success?

» Are you engaging your owner?

» Are you engaging your most expensive resource – your staff
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/rsenr/nr385se/mod3/complexity.html
Complex Ecosystems
Traditional workflow and staffing

» Waterfall
  » Does have Quality Assurance within a phase
  » It’s not incremental
  » Its not iterative between phases
Traditional Workflow and Staffing

The Architects Handbook of Professional Practice
History of LEAN in construction/software

1992 Toyota Production System pub.
1996 Rapid Application Development
1997 Last Planner System, Toyota Way Published
2000 Managing Quality
2001 Target Value Design
2004 Teicholz Graph
2005 Managing Quality
2006 Agile Manifesto
2007 Teicholz graph shows no change
Programming + Target Value Delivery + Design

1. Client's Business Case
2. Sufficient to Price
3. Preliminary Cost
4. Colocation
5. Agile Charter
6. Expected Cost
7. Target Cost
8. Savings
9. Final Cost
10. Preliminary Design Programming
11. Assemble Team
12. Validate Cost
13. Target Value Design
14. Savings
15. Project Delivery Design
16. Production-Construction
17. Commissioning
18. Continuous Estimating = sprints
19. Est. GOALS
20. Collect FACTS
21. Test CONCEPTS
22. Determine NEED
23. State PROBLEM
24. "D" design
25. Savings
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.
An Agile workforce

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 & 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.
1. Early and continuous delivery
2. Welcome changing requirements, even late in development.
3. Deliver working software frequently.
4. People must work together daily throughout the project.
5. Motivated individuals…trust them to get the job done.
6. Convey information face-to-face.
7. Working software is the primary measure of progress.
8. Promote sustainable development. All should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design.
10. Simplicity—the art of maximizing the amount of work not done, is essential.
11. Self-organizing teams.
12. At regular intervals, the team reflects then tunes and adjusts.
1. Early and continuous delivery
2. Welcome changing requirements, even late in development.
3. Deliver working software frequently.
4. **People must work together daily throughout the project.**
5. **Motivated individuals...trust them to get the job done.**
6. **Convey information face-to-face.**
7. Working software is the primary measure of progress.
8. Promote sustainable development. All should be able to maintain a constant pace indefinitely.
9. **Continuous attention to technical excellence and good design.**
10. Simplicity—the art of maximizing the amount of work not done, is essential.
11. Self-organizing teams.
12. At regular intervals, the team reflects then tunes and adjusts.
1. Early and continuous delivery
2. Welcome changing requirements, even late in development.
3. Deliver working software frequently.
4. People must work together daily throughout the project.
5. Motivated individuals…trust them to get the job done.
6. Convey information face-to-face.
7. Working software is the primary measure of progress.
8. Promote sustainable development. All should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design.
10. Simplicity—the art of maximizing the amount of work not done, is essential.
11. Self-organizing teams.
12. At regular intervals, the team reflects then tunes and adjusts.
The State of Agile

» **Who’s using it:** Tech, Financial services, Healthcare …we are “other” at the very bottom of the list.

» **Why Adopt it?** Accelerate delivery, ability to manage changing priorities, Increase productivity

» **Top Five Techniques:** Daily standup, sprint planning, retrospectives, sprint review, short iterations

» **Leading causes of failure:** company culture, lack of experience, lack of management support.

*Deltek + AVITRU*
Kanban

» literally “card you can see;” sign or billboard
  » Visualize the workflow (Plan)
  » Limit Work in Progress (WIP)
  » Measure / Manage flow
  » Make policies explicit
  » Implement feedback loops
  » Improve collaboratively, evolve experimentally

It is not a software or management technique. It is a method for improving process that came from manufacturing.

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.
Scrum – User Stories

» Each item, or story, in the product backlog should include the following information:

  » Who - it is for
  » What - needs to be built
  » Why - we should do it: Owner Value and cost
  » An estimate of work to do
  » Acceptance criteria to help us know when it’s done (and correct) Build the test first!

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
Cartoon set as backlog?

- Describe scope
  - Location
  - User Story
  - CoA (QA checklist?)
Design
Scrum\Kanban your way to Key Performance Indicators

» Profitability:
» Net revenue:
» Average collection period:
» Multipliers:
» Average billing rate:
» Estimate to complete:
» Effective billing rate:
» Cost variance:
» Estimate at complete:
» Client satisfaction:
» On-time delivery:
» Schedule variance:
» Earned value management:
Scrum\Kanban your way to Key Performance Indicators

» Profitability:
» Net revenue:
» Average collection period:
» Multipliers:
» Average billing rate:
» Estimate to complete:
» Effective billing rate:
» Cost variance:
» Estimate at complete:
» Client satisfaction:
» On-time delivery:
» Schedule variance:
» Earned value management:
Scrum\Kanban your way to Key Performance Indicators

» Estimate to complete (ETC) Measure either $ or % needed to complete.

» Scrum: As each sprint is competed, the teams “velocity” (completed work {points} in a given period) is known. This allows estimating: work done – work to do = cost or % of remaining work.

» Kanban: Strictly speaking there is no due date in Kanban, it’s just a prioritized list and WIP limits.

» By itself kanban does not give us the ETC however it can help in developing our next point: Estimate at Completion (EAC)

» Burn down Chart: good in either case
Kanban your way to Key Performance Indicators

» Estimate at completion (EAC):
  » Scrum or Kanban: We have the Actual Cost (AC) in the form of the actual work done to date. We can then add this to the Budget at Completion (BAC) minus the Earned Value (EV)
  » EAC=AC+(BAC-EV)
  » Wow! is that it? Well it is if you assume that the money spent and product complete are both the same (e.g. 40% of the money and 40% of the work)
  » Are they? If not we need to estimate Earned Value.
Kanban your way to Key Performance Indicators

Earned value management:

- You need:
  - a plan (project backlog)
  - Valuation of planned work (story points?)
  - “earning rules” (conditions of acceptance)

Kanban your way to Key Performance Indicators

» Cost variance: completed cost compared to planned cost

» Scrum or Kanban:
  » Compare your plan to what was accomplished.
  » Both techniques provide an explicit record of completed work.
  » Translate story points or other work estimates into dollars by multiplying “average billing rate” by the staff time used as compared to the contract amounts.
Kanban your way to Key Performance Indicators

» Schedule variance:
  » Scrum or Kanban:
    » Compare your plan to what was accomplished.
    » Both techniques provide an explicit record of completed work.
  » Look at the rate of finished tasks.
  » Use a burn down chart!

Chart: https://en.wikipedia.org/wiki/Burn_down_chart
Kanban your way to Key Performance Indicators

» Client satisfaction:
  » Engage your clients regularly
  » Make them part of your process
  » Does this require more of their time? Probably.
  » Is it worth it? Arguably… yes.
  » Setting expectations and continual review and check-in can only help a relationship (ask your spouse)
Kanban your way to Key Performance Indicators

» On-time delivery:
  » In a “pull system” where “delivery” is triggered by an event …the connection is tenuous.
  » Your attempt is to keep your team at a “jogging” pace forever not a “run and drop exhausted” pace.
  » The “State of Agile report”* found that the primary advantages to agile are that it can:
    » Accelerate delivery,
    » Has the ability to manage changing priorities,
    » Increases productivity

* [https://www.stateofagile.com](https://www.stateofagile.com)
The Future
Develop a DevOps Culture

https://www.ayaval.com/

Deltek + AVITRU

http://imaginexconsulting.com/devops-war-stories/ and
https://www.slideshare.net/devopsdaysaustin/2016-ignite-devops-or-noops
Develop a DevOps Culture

**DevOps**: refers to a set of practices that emphasize the collaboration and communication of both software developers and information technology (IT) professionals while automating the process of software delivery and infrastructure changes. It aims at establishing a culture and environment where building, testing, and releasing software can happen rapidly, frequently, and more reliably.

Develop a DevOps Culture

**DevOps**: refers to a set of practices that emphasize the collaboration and communication of both Designers and Specifiers, Estimators and Construction Professionals while automating the process of Construction Project delivery and infrastructure changes.

Infrastructure changes: including the software and processes used in model and data creation and maintenance throughout the buildings life

It aims at establishing a culture and environment where building, testing, and releasing design and construction workface packages can happen rapidly frequently, and more reliably.

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 from it tomorrow. If it was not realistic or if the team bit off more than it could chew, this is feedback to improve planning.
“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
Risk reduction

» Using these concepts:
  » Do work!
  » Gather and analyze Data about that work
  » Rinse and Repeat
Questions?

If you have questions about this presentation please contact me at:

» Marcchavez@Deltek.com