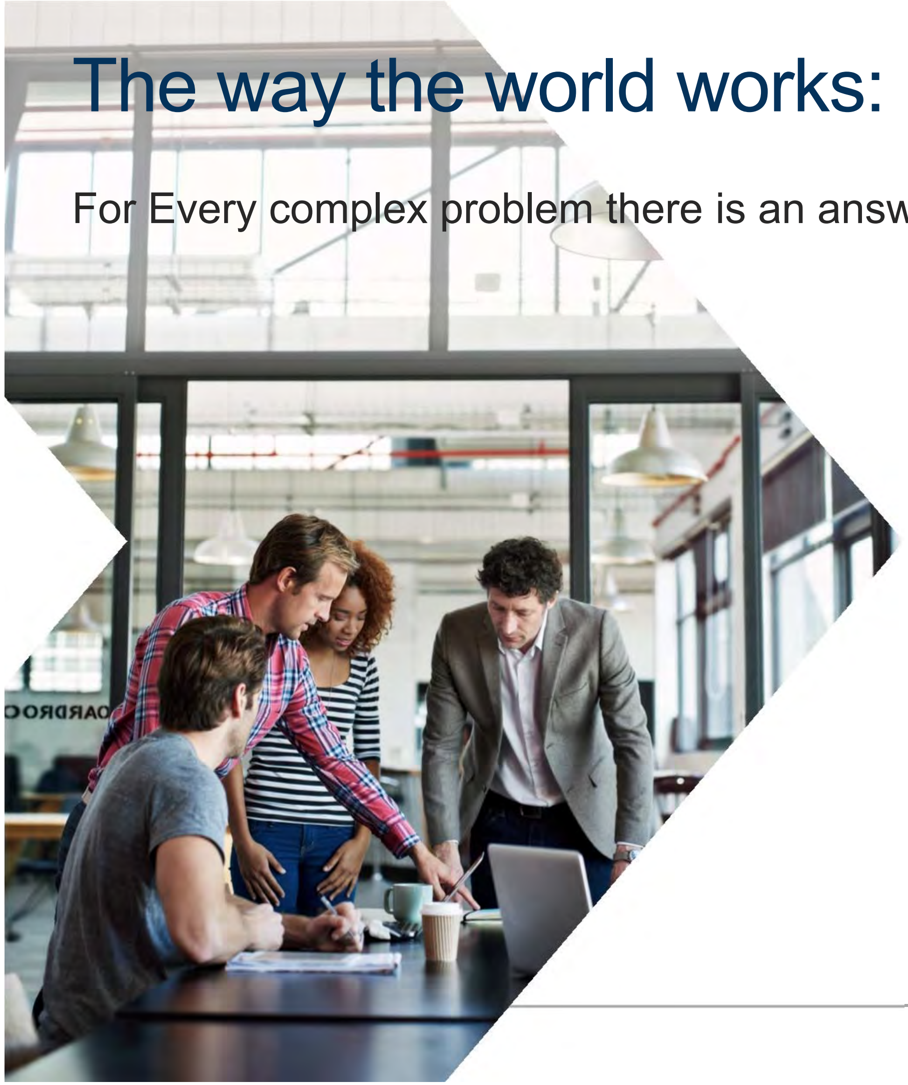
Deltek + AVITRU

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

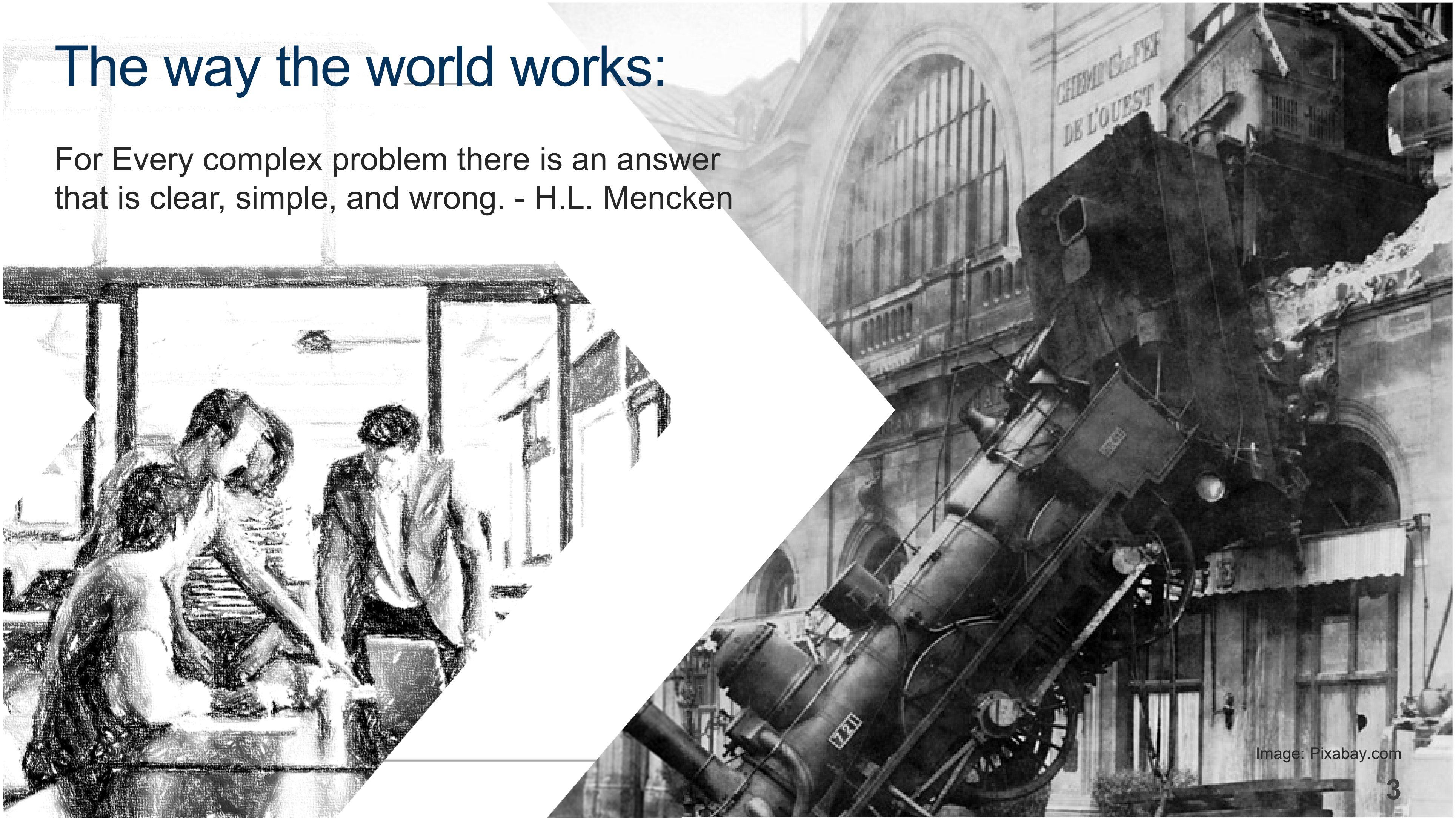


Image: Pixabay.com

Deltek + AVITRU

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

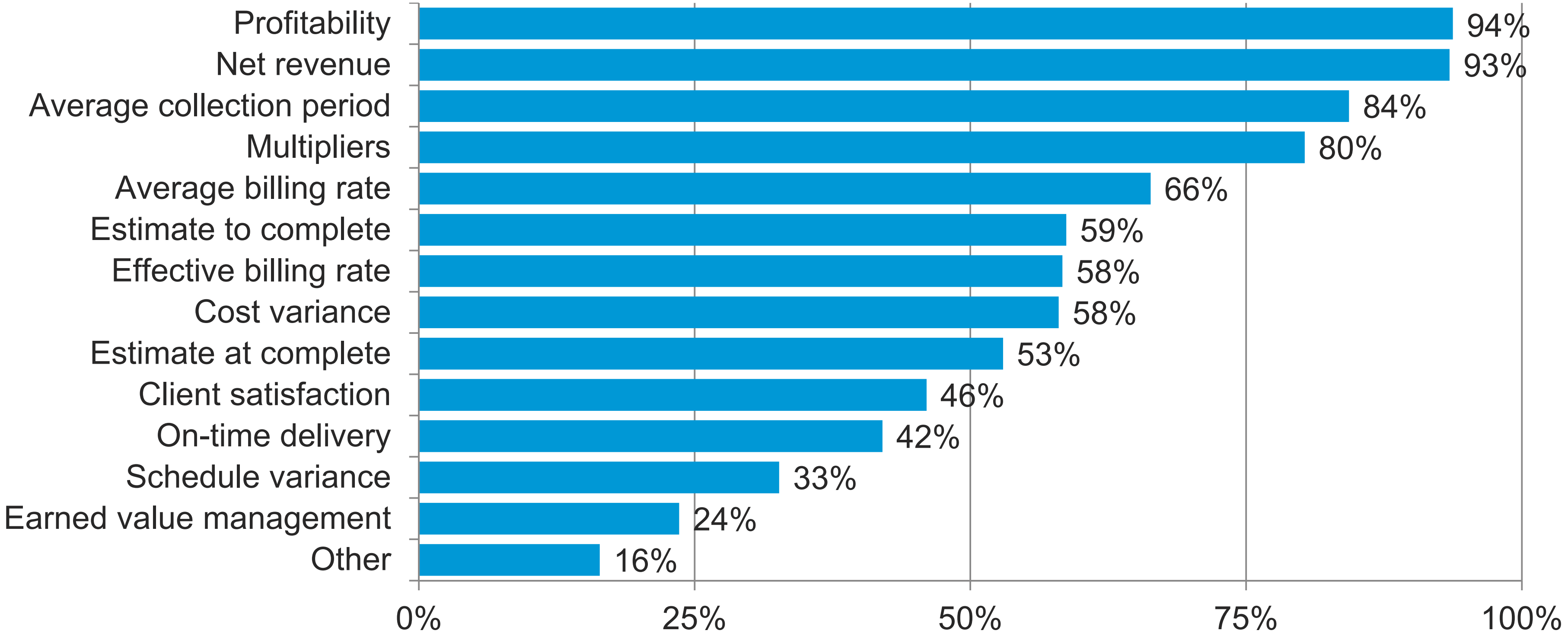
Of collected data in 2018 175 ZB in 2025.

dead last

Amount of data stored per firm >1000 employees in 2009 (231 terabytes) “professional services” = 278

.

Project Management KPIs Tracked



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%)

Projects On or Under Budget

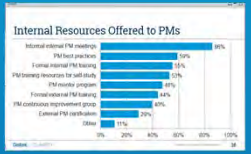
Projects On or Ahead of Schedule

Firms Measuring Client Satisfaction

Internal Project Performance Evaluations

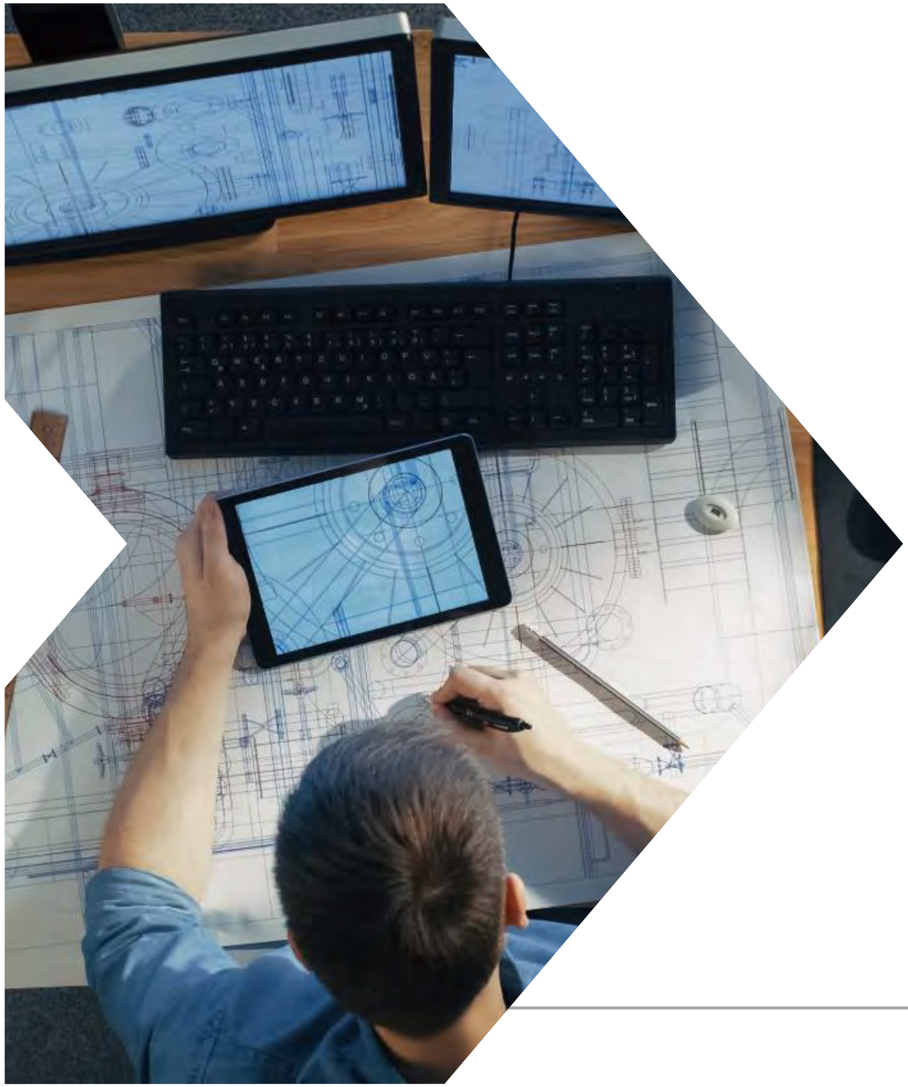
Projects Using Clearly Defined PM Process

Project Leaders with Formal Project Management Training


Internal Resources Offered to PMs

Top Project Management Initiatives

Deltek | CLARITY

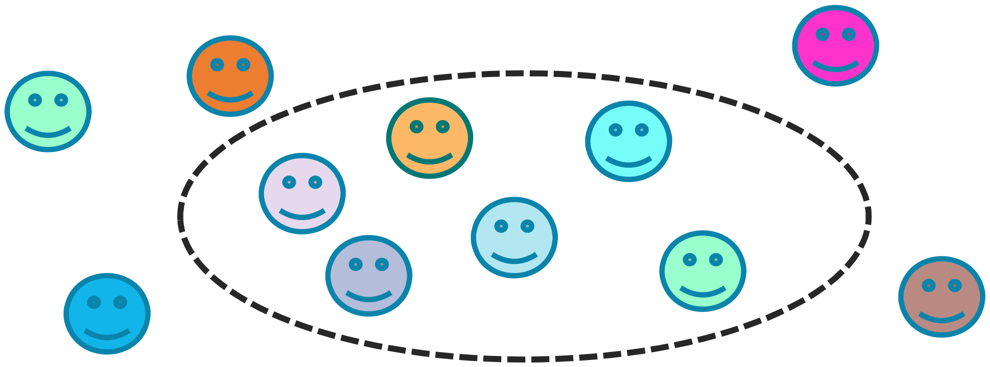


Project Management: Next steps

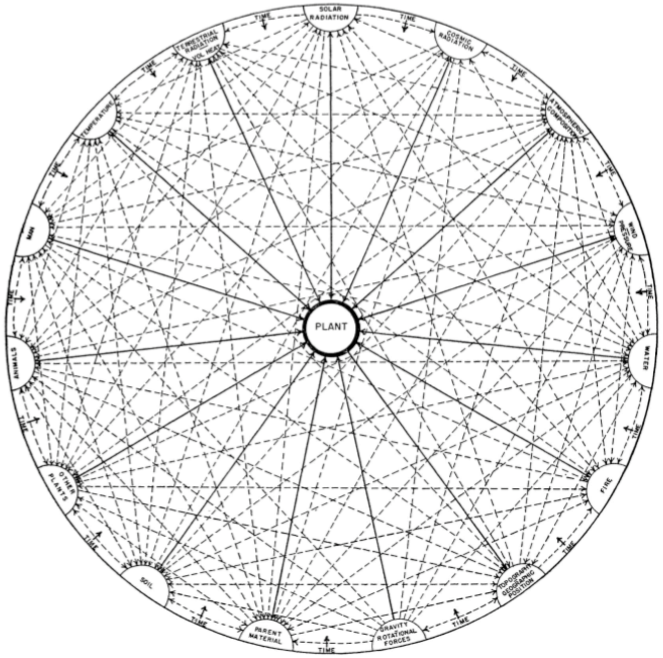
- What’s holding your projects back?
- What do PMs need to more proactively manage projects?
- Are you providing construction feedback to your team?
- Are your processes hindering your success?
- Are you engaging your owner?
- Are you engaging your most expensive resource – your staff

Data Gathering and Analysis

- Construction is a COMPLEX SYSTEM we need data to analyze so we can improve our process

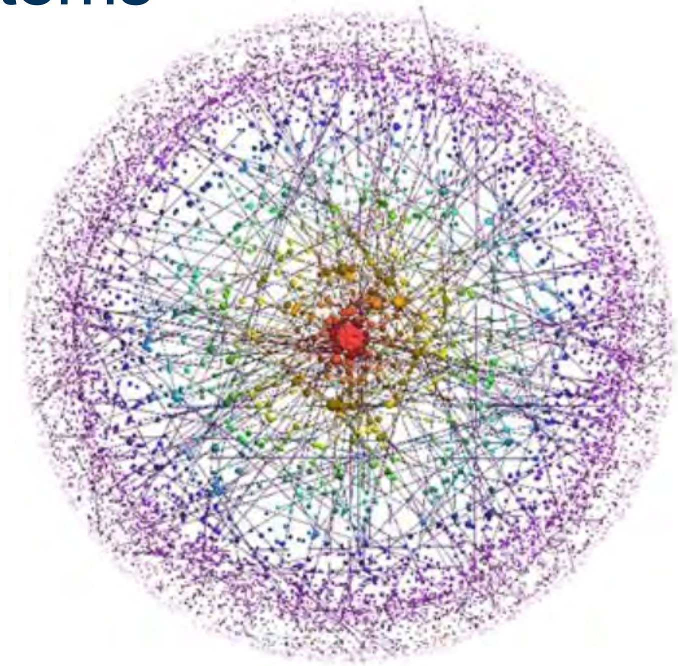


Complex Ecosystems



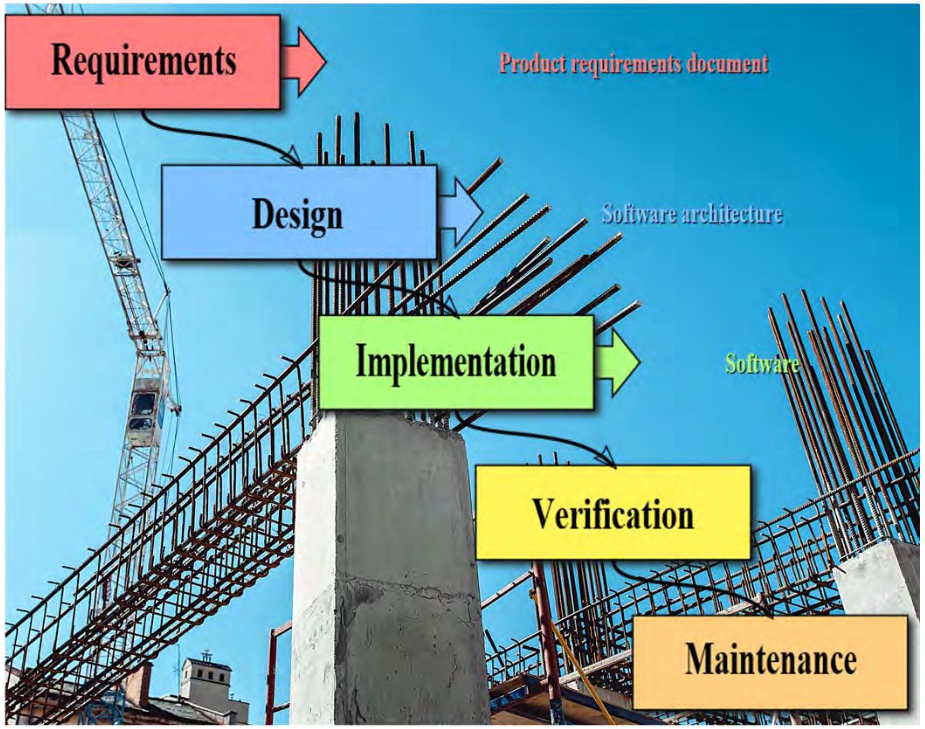
http://www.uvm.edu/rsenr/nr385se/mod3/complexity.html

Complex Ecosystems



http://www.lnsresearch.com/research-library/research-articles/managing-the-complex-supply-chain-ecosystem

Traditional workflow and staffing

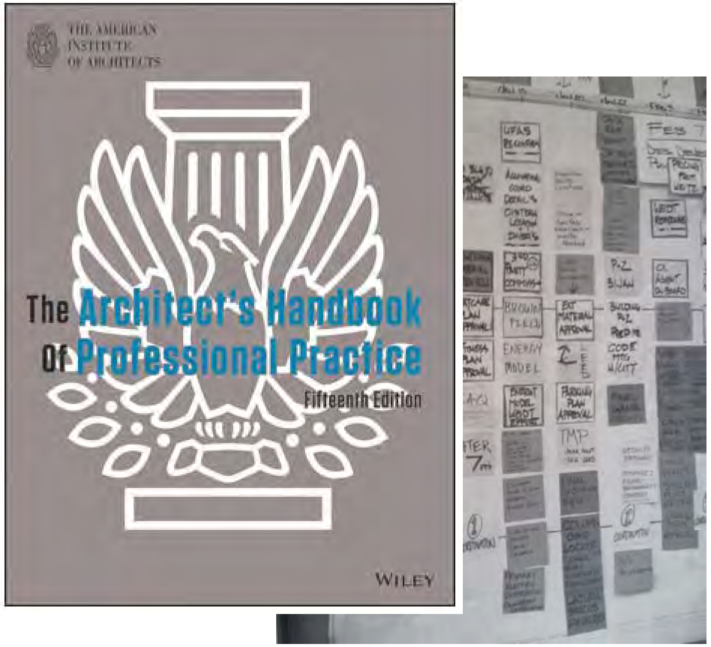


- Waterfall
  - Does have Quality Assurance within a phase
    - It’s not incremental
    - Its not iterative between phases

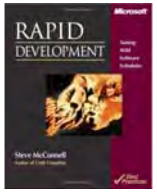
By Peter Kemp / Paul Smith - Adapted from Paul Smith's work at Wikipedia, CC BY 3.0,
https://commons.wikimedia.org/w/index.php?curid=10633070

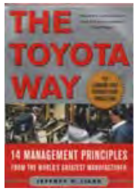
Traditional Workflow and Staffing

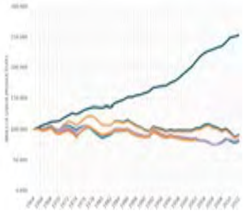
The Architects Handbook of Professional Practice

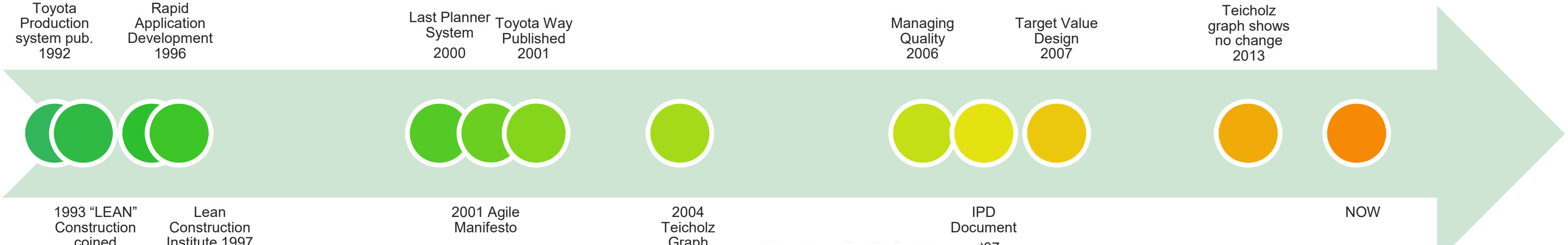

THE AMERICAN INSTITUTE OF ARCHITECTS
The Architect's Handbook Of Professional Practice
Fifteenth Edition
WILEY

History of LEAN in construction/software









Toyota Production system pub. 1992

Rapid Application Development 1996

Last Planner System 2000

Toyota Way Published 2001

Managing Quality 2006

Target Value Design 2007

Teicholz graph shows no change 2013

1993 “LEAN” Construction coined

Lean Construction Institute 1997

2001 Agile Manifesto

2004 Teicholz Graph

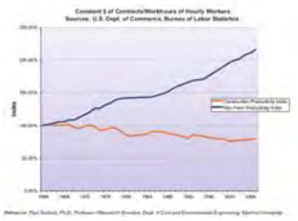
IPD Document)07

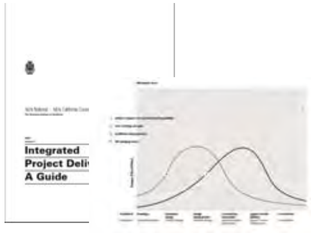
NOW

Lean Construction Institute
Transforming Design and Construction

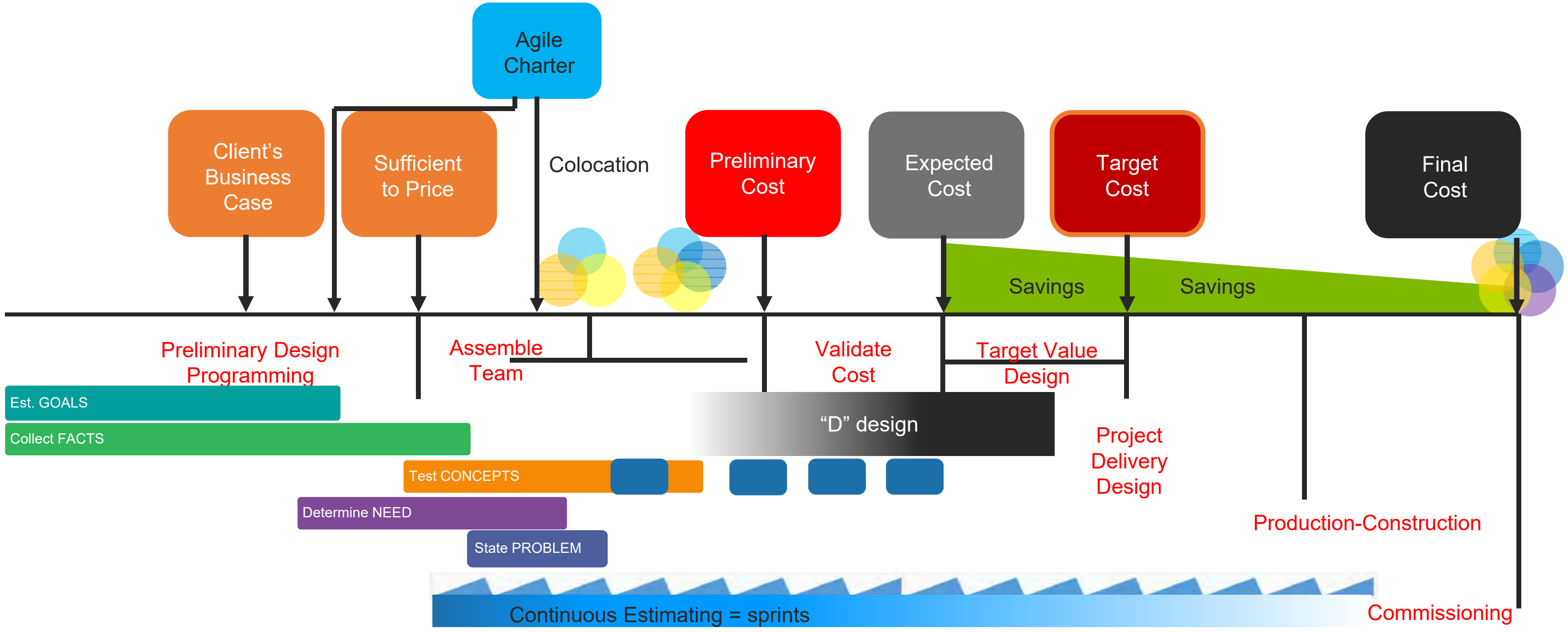
Manifesto for Agile Software Development

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





Programming + Target Value Delivery + Design



Deltek + AVITRU

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.

Agile

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.

Agile

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.

Agile

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.

Kanban

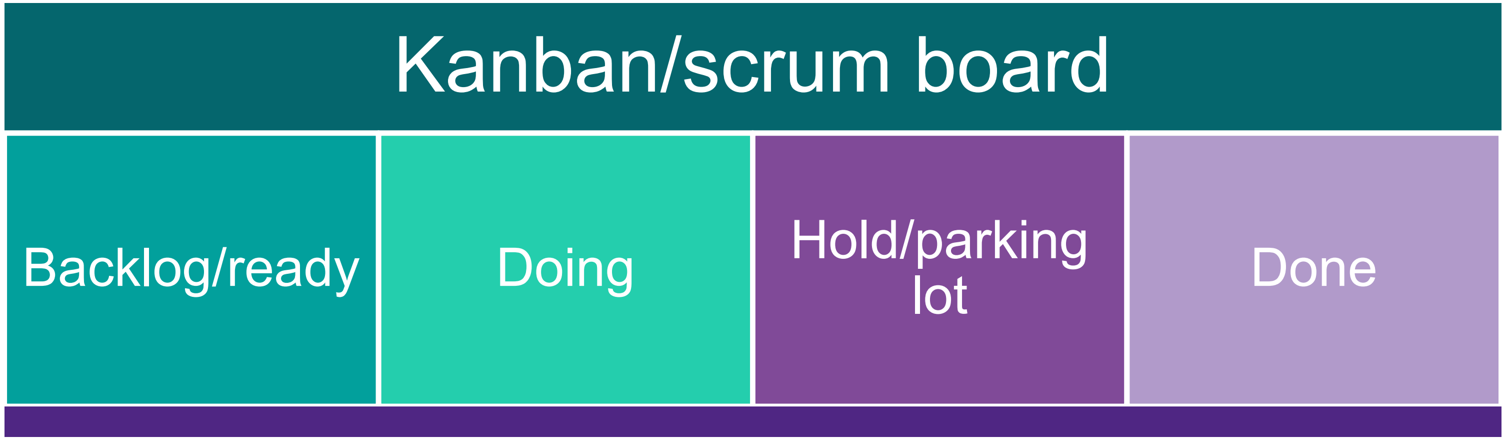
- literally “card you can see;” sign or billboard
  - Visualize the workflow (Plan)
  - Limit Work in Progress (WIP)
  - Measure / Manage flow
  - Make policies explicit
  - Implement feedback loops
  - Improve collaboratively, evolve experimentally

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.



Kanban/scrum board

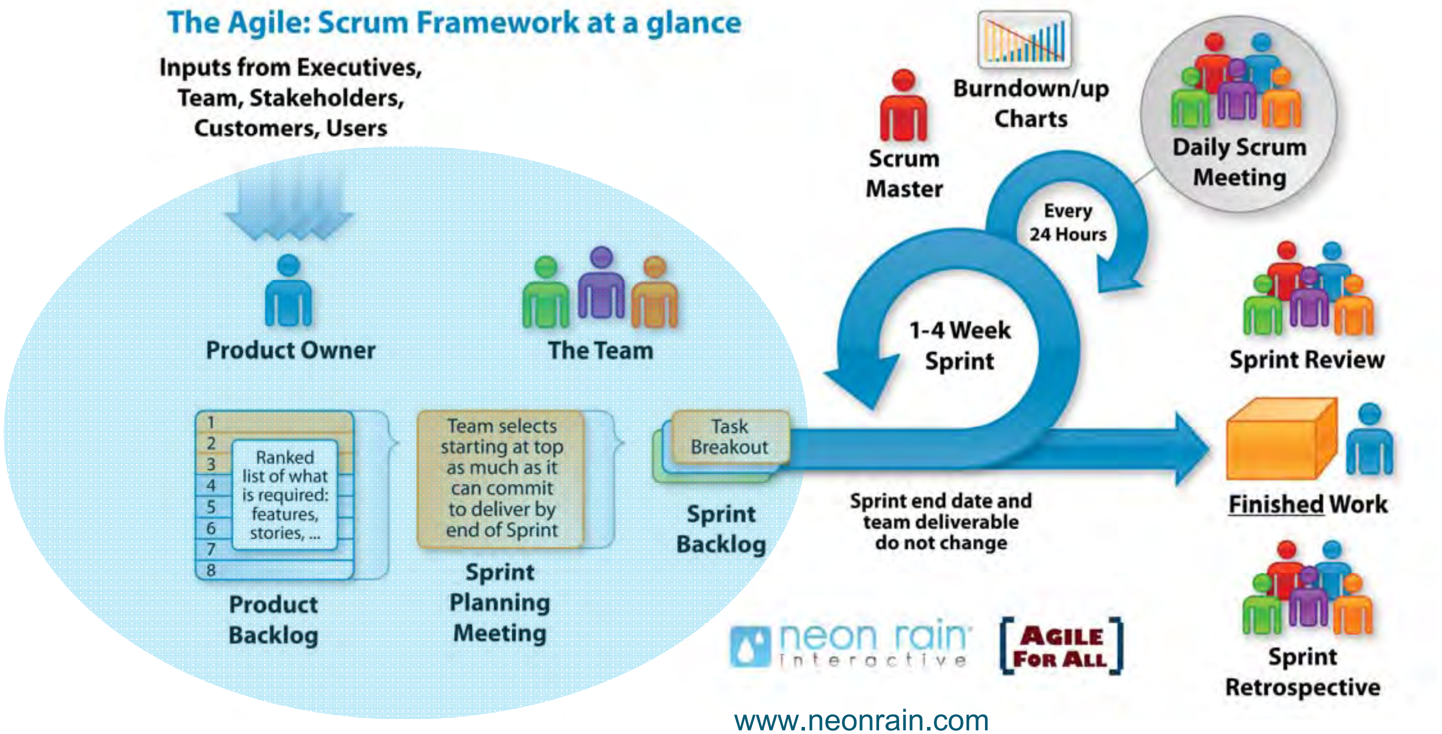
Backlog/ready

Doing

Hold/parking lot

Done

Scrum


The Agile: Scrum Framework at a glance
Inputs from Executives, Team, Stakeholders, Customers, Users
Product Owner
The Team
1
2
3
4
5
6
7
8
Ranked list of what is required: features, stories, ...
Product Backlog
Team selects starting at top as much as it can commit to deliver by end of Sprint
Sprint Planning Meeting
Task Breakout
Sprint Backlog
Scrum Master
Burndown/up Charts
Daily Scrum Meeting
Every 24 Hours
1-4 Week Sprint
Sprint end date and team deliverable do not change
Sprint Review
Finished Work
Sprint Retrospective
neon rain interactive
Agile For All

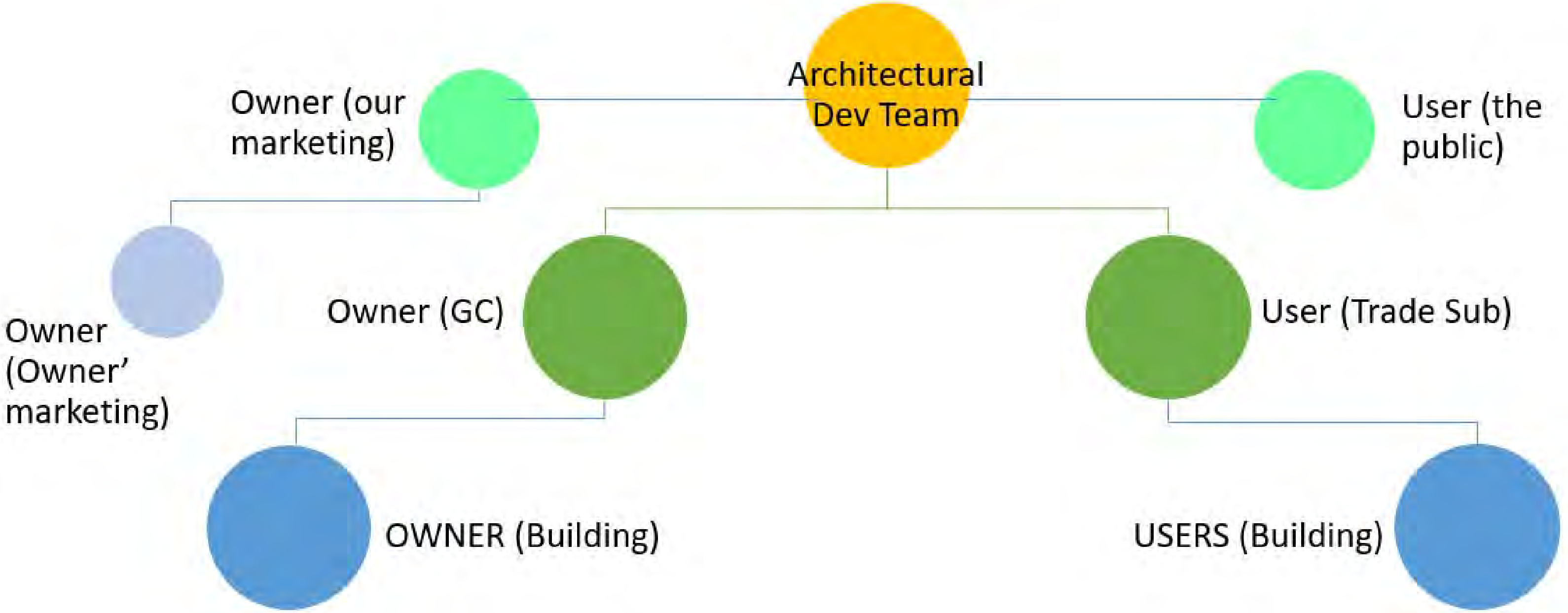
www.neonrain.com

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!

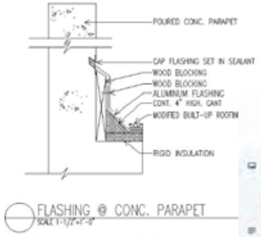
As a .... <*type of user*>
I want to .... <do something> (*some business action)*
Because I need to .... <some value created> (the reason why we did it).

Architectural practice as an Intermediate

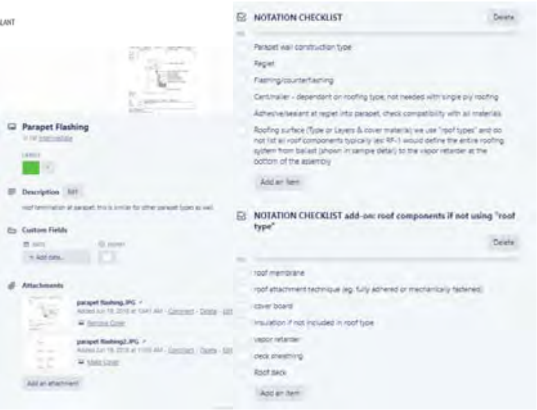


Cartoon set as backlog ?

- Describe scope
    - Location
    - User Story
    - CoA (QA checklist?)



FLASHING @ CONC. PARAPET
SCALE 1-1/2"=1'-0"



Parapet Flashing

LABELS

🟩

Description Edit

roof termination at parapet. this is similar for other parapet types as well.

Custom Fields

+ Add date...

Attachments

parapet flashing.JPG
Added Jun 18, 2018 at 10:41 AM - Comment - Delete - Edit
Remove Cover

parapet flashing2.JPG
Added Jun 18, 2018 at 11:00 AM - Comment - Delete - Edit
Make Cover

Add an attachment

NOTATION CHECKLIST Delete

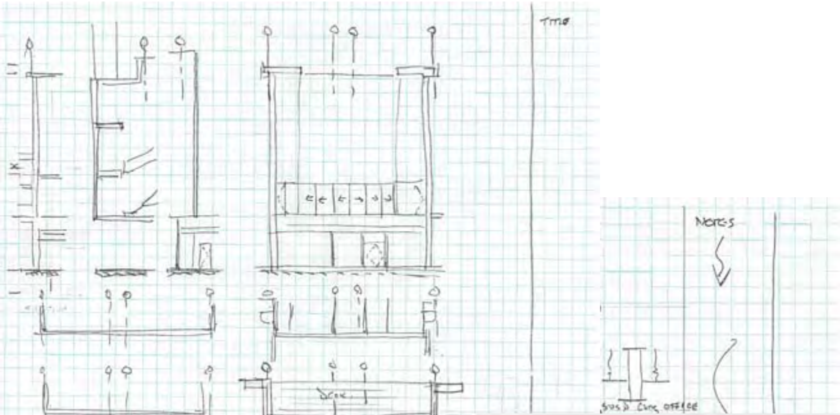
- Parapet wall construction type
- Reglet
- Flashing/counterflashing
- Cant/nailer - dependant on roofing type; not needed with single ply roofing
- Adhesive/sealant at reglet into parapet, check compatibility with all materials
- Roofing surface (Type or Layers & cover material) we use "roof types" and do not list all roof components typically (ex: RF-1 would define the entire roofing system from ballast (shown in sample detail) to the vapor retarder at the bottom of the assembly

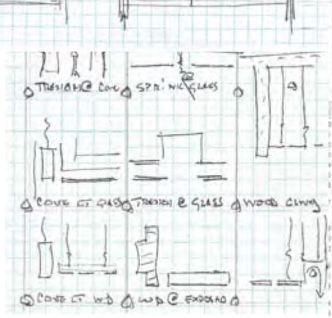
Add an item

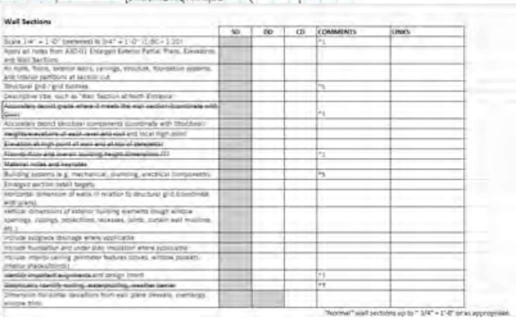
NOTATION CHECKLIST add-on: roof components if not using "roof type" Delete

- roof membrane
- roof attachment technique (eg. fully adhered or mechanically fastened)
- cover board
- insulation if not included in roof type
- vapor retarder
- deck sheathing
- Roof deck

Add an item







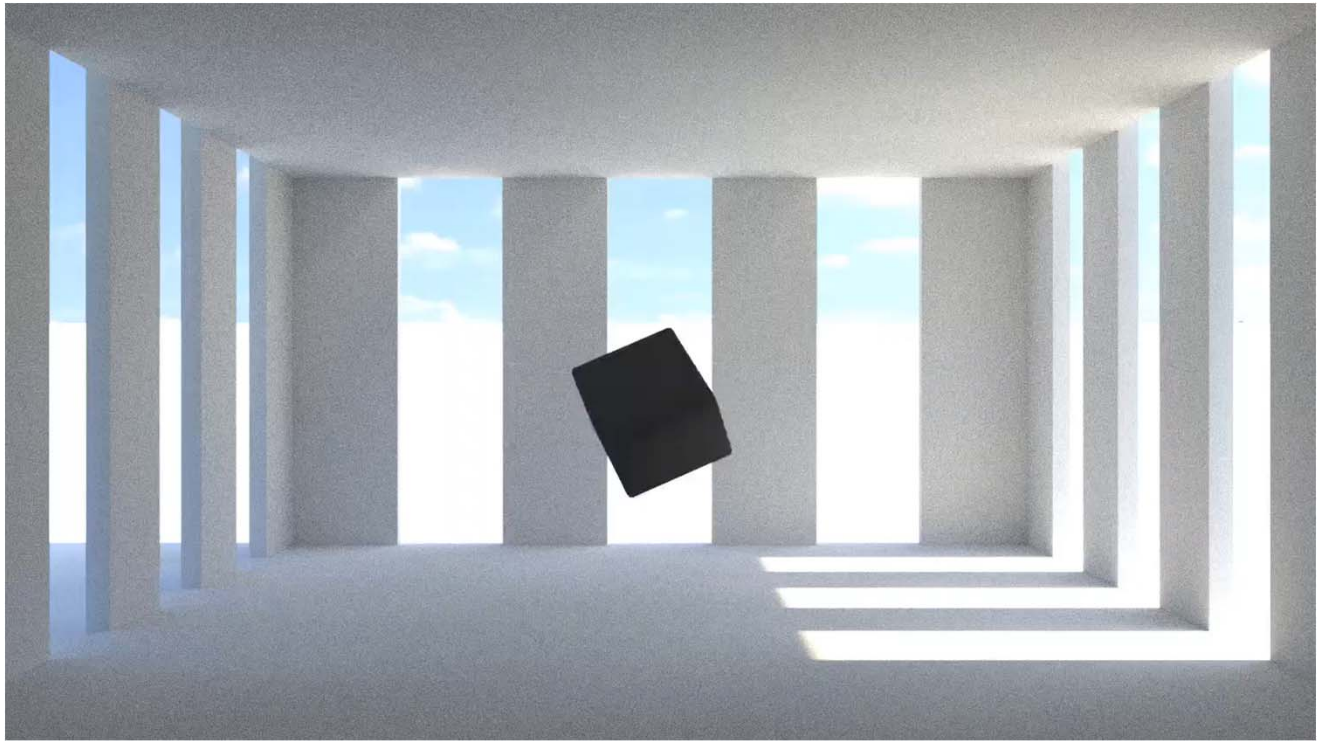
Wall Sections

	SD	DD	CD	COMMENTS	LINKS
Scale 1/4" = 1'-0" (preferred) to 3/4" = 1'-0" (1:50 – 1:20)				*1	
Apply all notes from A3D-01 Enlarged Exterior Partial Plans, Elevations and Wall Sections					
All roofs, floors, exterior walls, canopies, structure, foundation systems, and interior partitions at section cut					
Structural grid / grid bubbles				*1	
Descriptive title, such as "Wall Section at North Entrance"					
~~Accurately depict grade where it meets the wall section (coordinate with Civil)~~				*1	
Accurately depict structural components (coordinate with Structural)					
~~Heights/elevations of each level and roof~~ and local high point					
~~Elevation at high point of wall and at top of parapet(s)~~					
~~Floor-to-floor and overall building height dimensions (?)~~				*1	
~~Material notes and keynotes~~					
Building systems (e.g. mechanical, plumbing, electrical components)				*5	
Enlarged section detail targets					
Horizontal dimension of walls in relation to structural grid (coordinate with plans)					
Vertical dimensions of exterior building elements (rough window openings, ceilings, projections, recesses, joints, curtain wall mullions, etc.)					
Include subgrade drainage where applicable					
Include foundation and under slab insulation where applicable					
Include interior ceiling perimeter features (coves, window pockets, shadow shades/blinds)					
~~Identify important alignments~~ and design intent				*1	
~~Graphically identify roofing, waterproofing, weather barrier~~				*?	
Dimension horizontal deviations from wall plane (reveals, overhangs, window trim)					

"Normal" wall sections up to " 3/4" = 1'-0" or as appropriate.

Deltek + AVITRU

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)

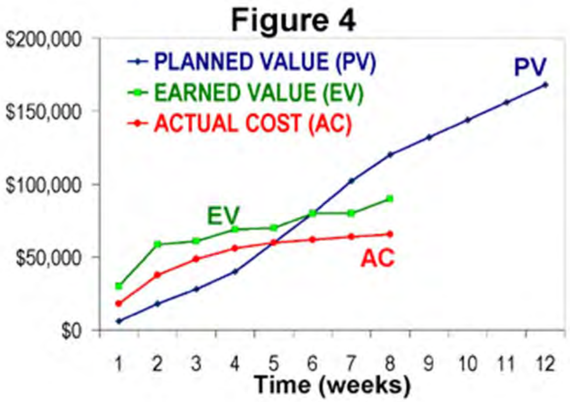


Chart: https://en.wikipedia.org/wiki/Earned_value_management

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!

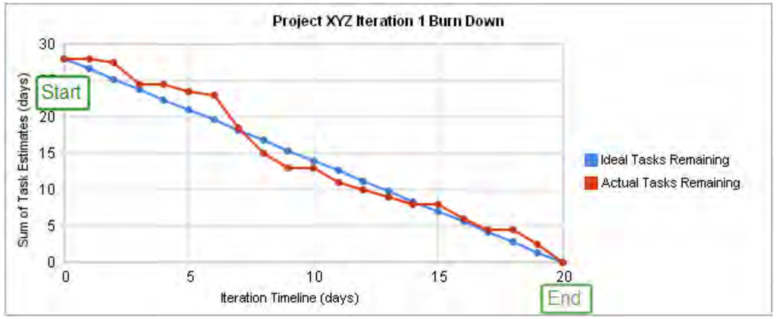


Chart: https://en.wikipedia.org/wiki/Burn_down_chart

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

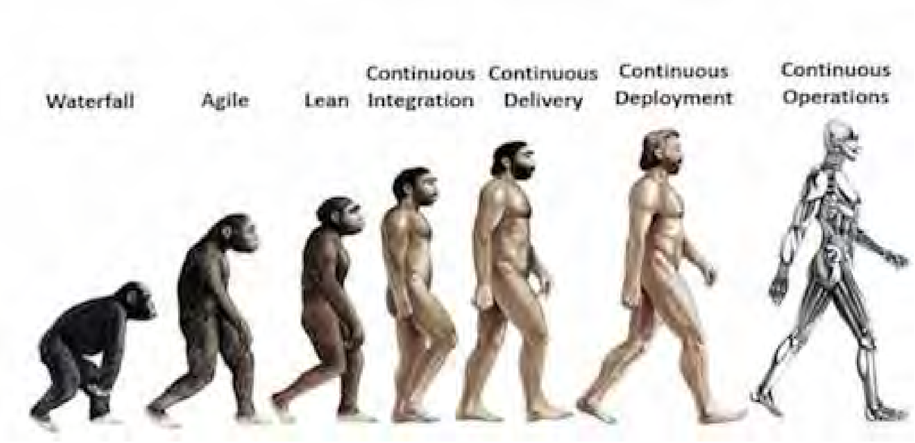
- On-time delivery:
  - In a “pull system” where “delivery” is triggered by an event …the connection is tenuous.
  - Your attempt is to keep your team at a “jogging” pace forever not a “run and drop exhausted” pace.
  - The “State of Agile report”* found that the primary advantages to agile are that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

* https://www.stateofagile.com

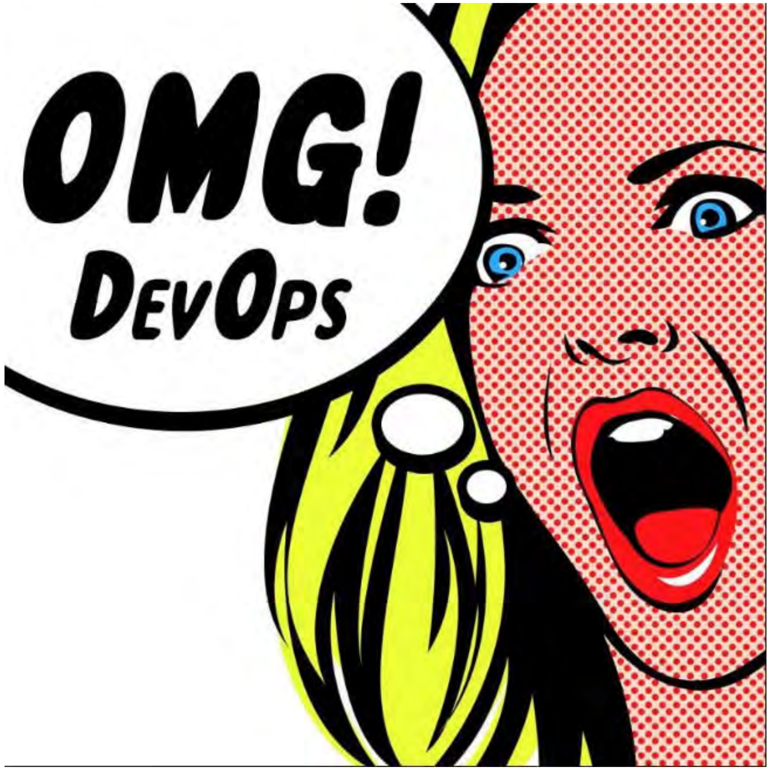
Deltek + AVITRU

The Future

Develop a DevOps Culture



https://www.ayaval.com/



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.

https://en.wikipedia.org/wiki/DevOps

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.

https://en.wikipedia.org/wiki/DevOps

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

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