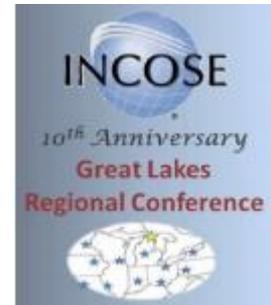


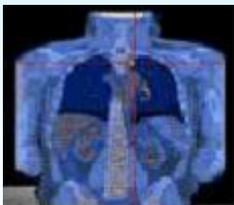
Agile at GEHC

Chris Unger, Chief Systems Engineer
Dave Duckert, Program Manager
Mikael Boman, Sr Staff Software Agile Leader,



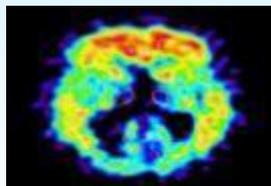
Broad Based Diagnostics

Diagnostic Imaging



- CT, PET/CT
- MR

Medical Diagnostics



- Contrast agents
- Molecular diagnostics

Clinical Systems



- Ultrasound
- Critical care systems

Information Technology & Services



- Electronic medical records
- Revenue cycle



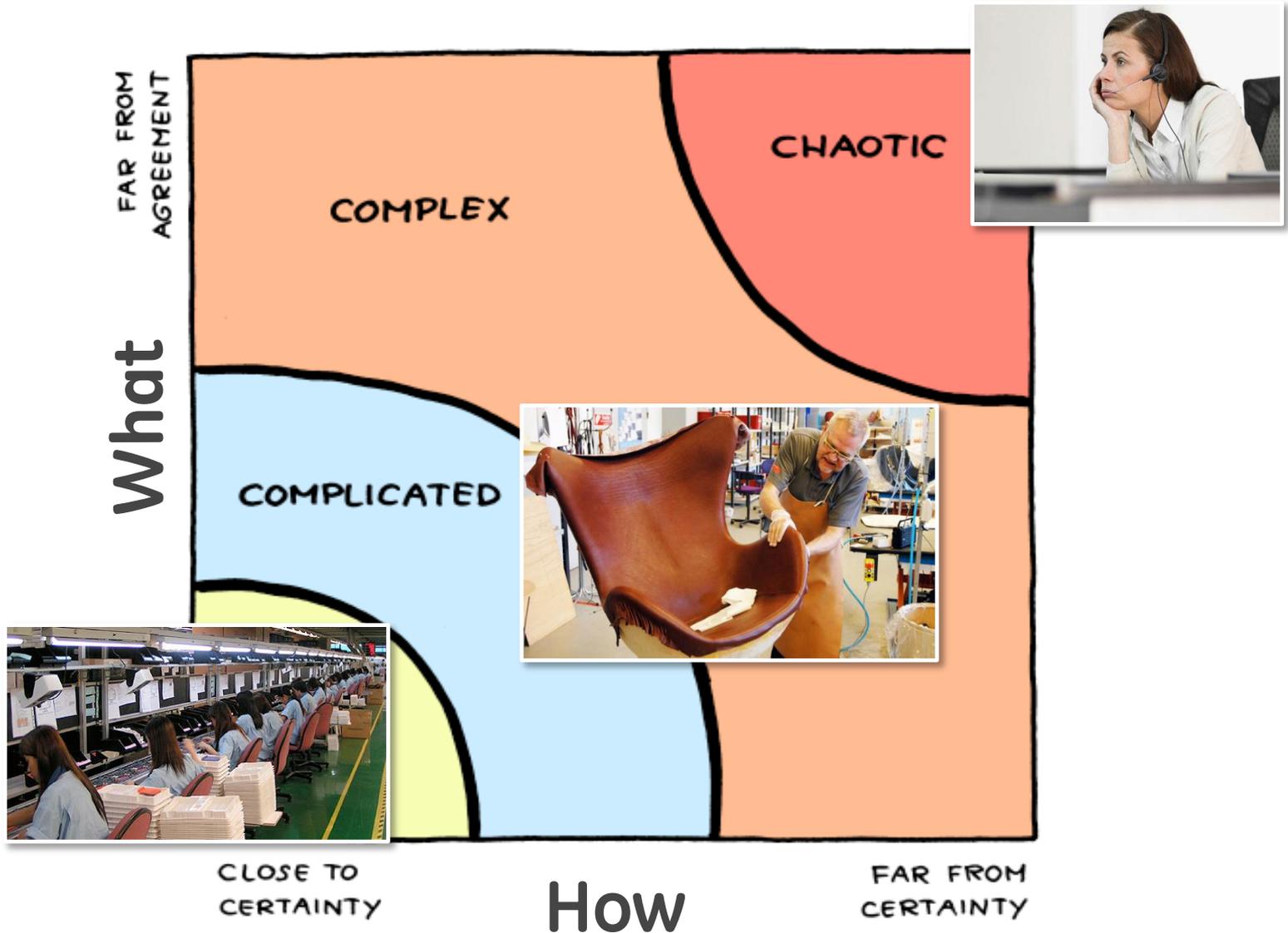
- Performance solutions
- Multi-vendor services

Life Sciences



- Discovery systems
- Protein separations

Manage Complicated and Complex problems



Barometer

There are many ways to achieve the same goal.

- Pressure
- Shadow
- Count
- Drop
- Trade

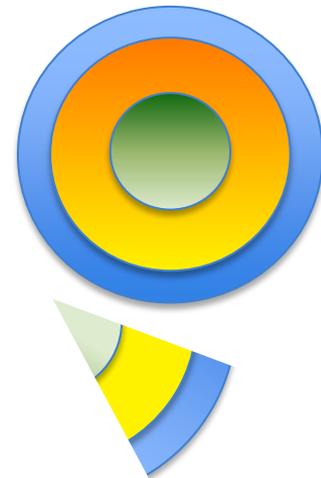


Alexander Calandra

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

Why Agile Software Development Methodology?

Program Challenges	Agile Strategies
<p>Quality</p> <ul style="list-style-type: none"> • Need continuous customer and performance feedback to ensure needs are met. 	<ul style="list-style-type: none"> • Frequent demo's, VoC. • Integrate cross functional team into the development process
<p>Predictability</p> <ul style="list-style-type: none"> • Better estimation tools • Fewer late execution surprises • Better progress measuring system 	<ul style="list-style-type: none"> • Decompose work into quarterly program increments...focus on customer expectations • Close monitoring and tracking of progress • <i>Automated testing</i> and "Definition of Done"
<p>Speed</p> <ul style="list-style-type: none"> • Need to re-prioritize quickly to address risks, support the hardware teams, test, VoC, etc. 	<ul style="list-style-type: none"> • Work is done in slices, not layers. • <i>Short 2 week sprints of focused activity to create a software development assembly line</i>
<p>Efficient and Scalable</p> <ul style="list-style-type: none"> • Better use of our resources • Need to partition the work to scale up the number of teams. • Integrated continuous improvement 	<ul style="list-style-type: none"> • Teams are self organizing and supported with dedicated execution, technical, and product functionality oversight.



What is an Agile way of working? (with all jargon gone!)

Part 1



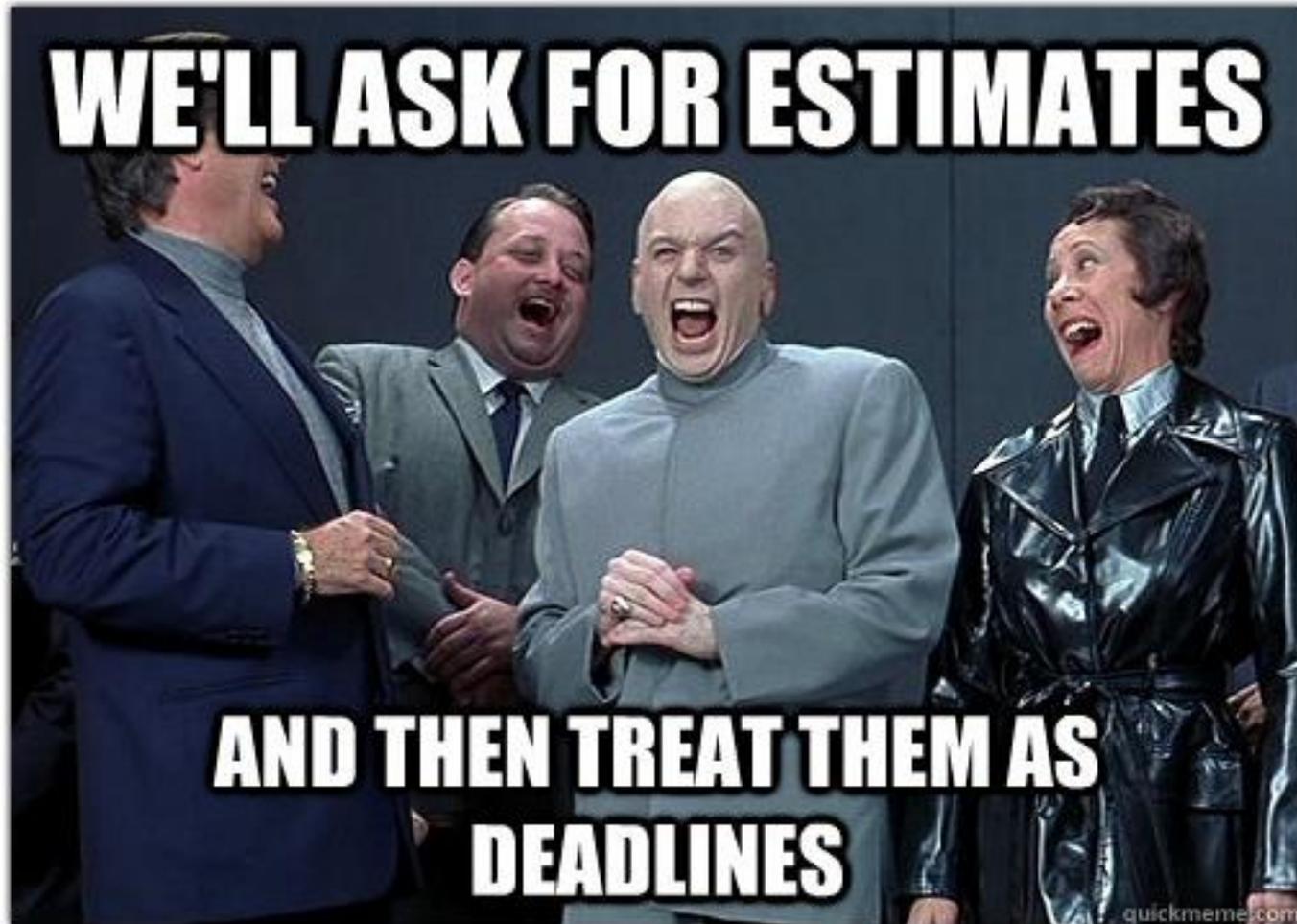
- ✓ Have a long term, prioritized plan for what you want to do
- ✓ Break down the work into smaller pieces and create a short term plan for the next few weeks
 - Make sure each task can be completed, and adds “value” (where value is defined by the “customer”) and minimize unplanned rework
 - Break the program team into independent teams that can complete valuable work
- ✓ Meet daily/often to share knowledge, support each other in making progress and identify & remove blockers
- ✓ At the end of the few weeks, stop and compare you plan for that period to what really happened – what is DONE?
- ✓ At the end of the few weeks, also stop to reflect on your way of working and challenge yourself to improve

What is an Agile way of working? (with all jargon gone!) Part 2



- ✓ Allow individuals and teams time to focus in order for them to be able to get things done
- ✓ Allow the relentless transparency to enable informed decision making
- ✓ If you are building something – try to form a team of all (or as many as is feasible) of the competencies needed to take the idea of what to build to something in the hands of the customers.
- ✓ Plan and execute for sustainable work levels...minimize short term surges to meet scope and time deadlines

Agile estimation – the problem we are trying to solve

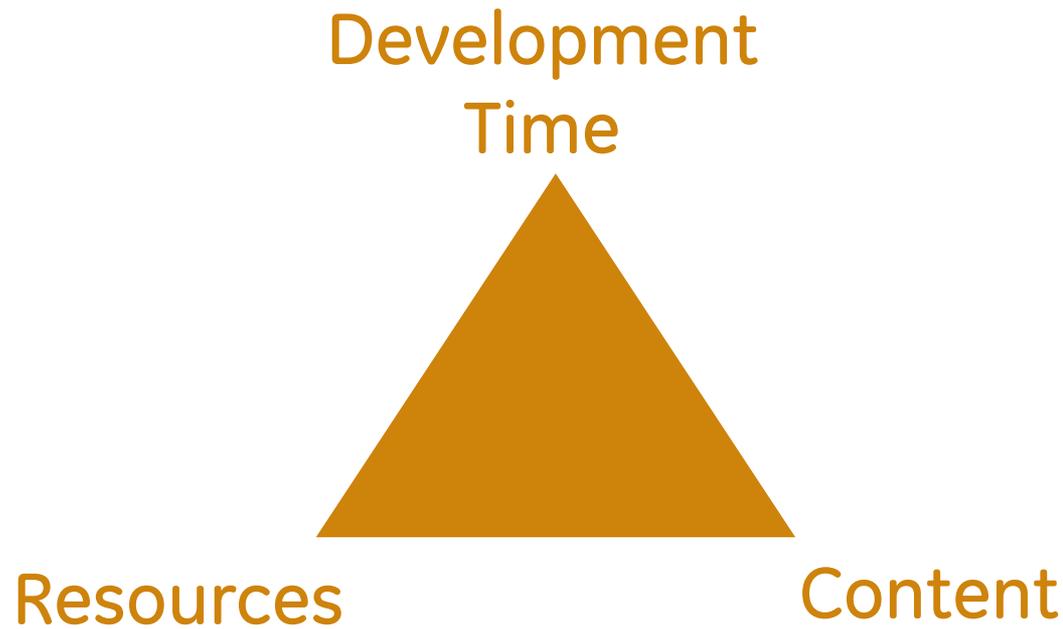


*It is better to be vaguely
right than precisely wrong*

- Carveth Read, British philosopher

Agile?

Agile fixes development intervals and allows content to scale...



Fixed end-date and fixed content is not Agile!

The GEHC Diagnostic Cardiology Portfolio

- Cardiac Cath Lab ▶
- EP Lab ▶
- ◀ Diagnostic ECG
- ◀ Echo Lab
- Cardiovascular Imaging ▶
- ◀ Administration



MAC 3500 MAC 2000



CASE



MAC 5500



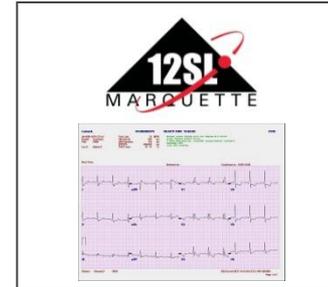
SEER Recorders



MARS Holter



MUSE



Algorithms

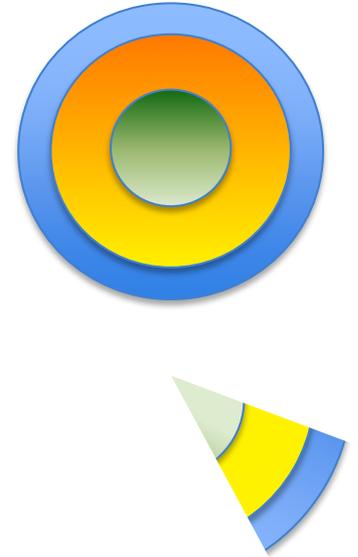


Remote Access

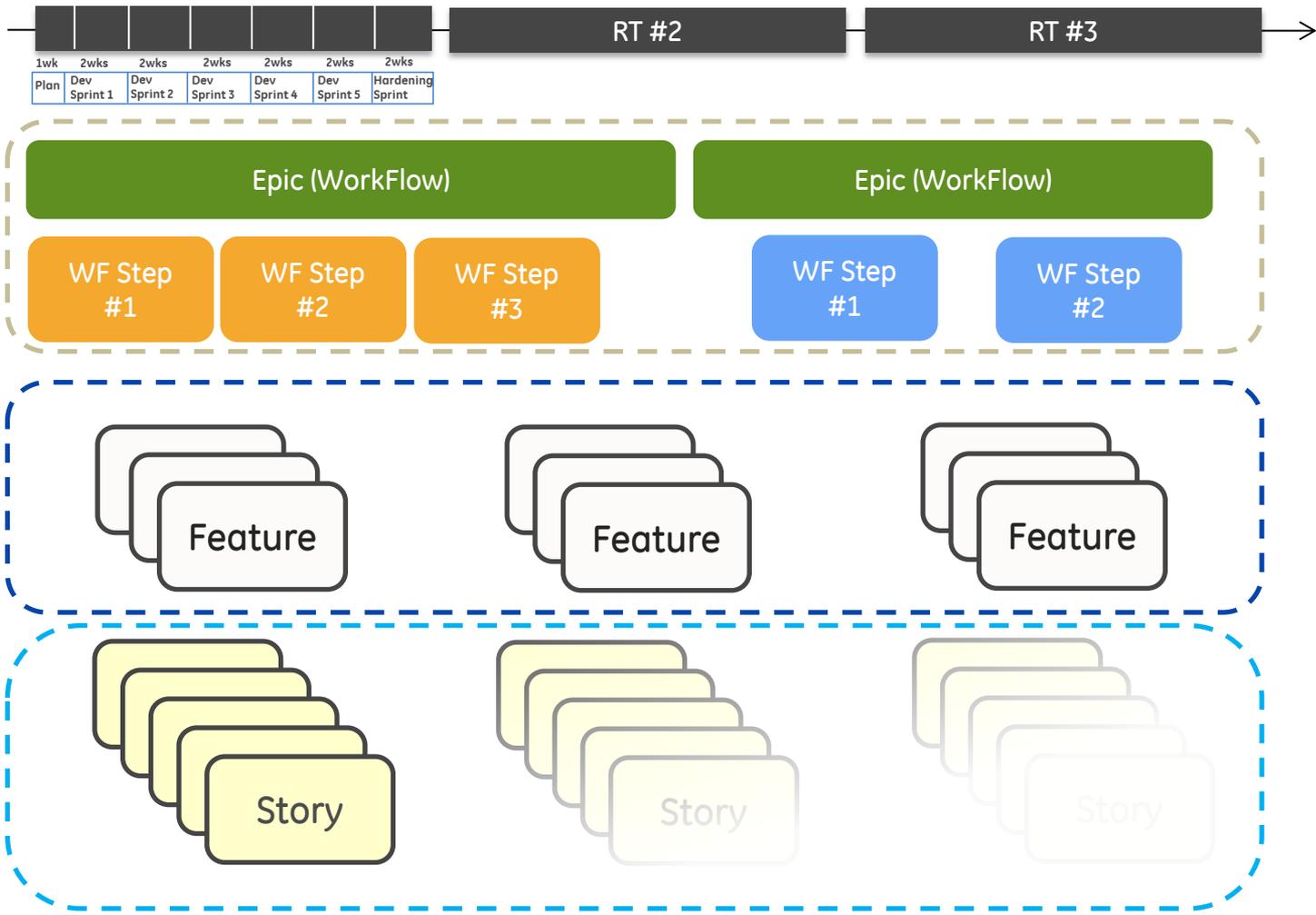
From cardiac acquisition to information at the doctor's fingertips...or wherever they want it

Deployment Example - Constraints

- New product/platform – slice model is challenging
- Product is hardware intensive - launch is complex and must be planned with suppliers, regulatory agencies, factories – end date must be known (roughly).
- ROI must be known!
- SW effort is capitalized – scope be known and fixed



Agile Software Development Timeline



>3 months
(Feature Points)

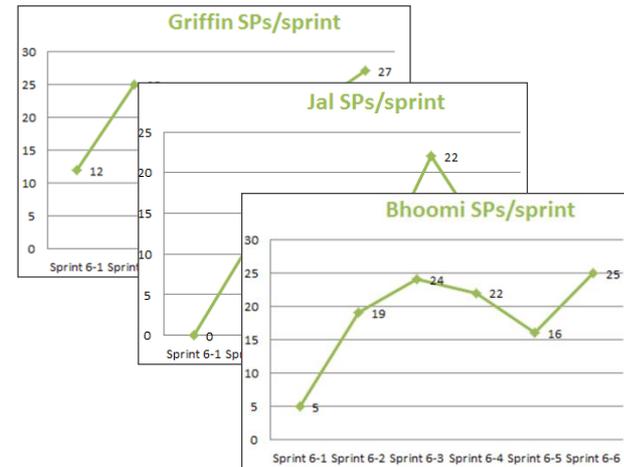
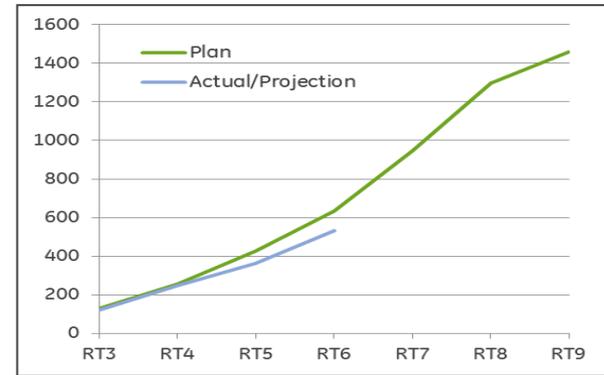
< 3months
FP=Feature Point

<2 weeks
SP=Story Point

*Hardware is Traditional Gantt Approach with SW Integration Points
Calendar-driven vs Effort Driven*

Program Tracking

- Progress against plan burn-up at FP level
- SP tracking at scrum team/sprint level
- Retros and feedback loop
 - Internal Benchmarking
 - Process Improvement (Acceleration)
 - Feature 'growth' (Deceleration)
 - Backlog grooming (Acceleration)



Sprint



Retro



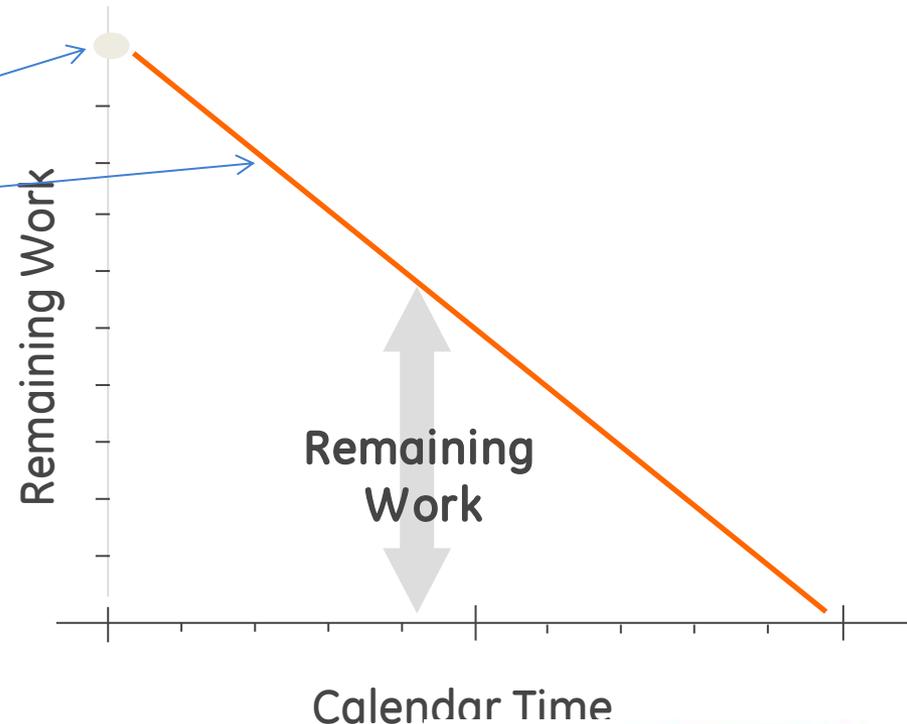
Root Cause: Changes to Processes, Roles, & Op Mechs

More Program Tracking

To predict the end date, we need...

1. Size of the Total Effort
2. Velocity

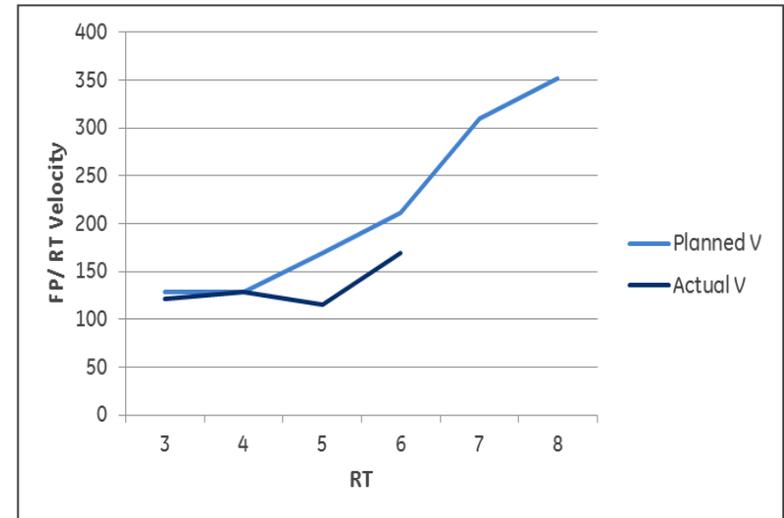
- Find the right granularity to make estimates (FP)
- Spend one quarter to measure velocity
- Extrapolate velocity over remaining work
- Integrate with larger program plan



Don't Forget: Ramp-up Time

New team members

- New teams dilute efforts of existing teams
- Allow 1Q ramp up – processes, tool chains, etc
- Perhaps add new members to existing teams rather than form new teams



Insights from the Data

- Velocity measure can be translated to \$/FP for outsourcing
- Compare internal/external team costs
- Burn up charts are early warning signs that something is wrong
- Improved credibility
- Improved Scope control
- Not punitive!



Challenges with Agile outside of software

Agile Scope and Requirements Management

- Tying scope to effort and market impact in an iterative/learning culture
- Incorporating lead times and technical risks into requirements phasing

Planning Culture and Practices

- Aligning with project governance framework - budgeting
- Getting away from the notion of “something functional every two/three weeks” being impossible in hardware/chemistry
 - Focus on a shared Definition of Done
 - Start thinking in new ways: 3D printing hardware, hardware interface first development, etc

Staffing Culture and Practices

- Distributed team members: Few programs are fully collocated
- Focus: Habit of manning people in multiple (5-10) projects at once

Harmonizing Terminology

- The principles of good engineering are consistent, but each (corporate) initiative has either different words and techniques for the same concept (bad) or uses the same word to mean slightly different things (worse)

Integrating Documentation: Agile, Compliant, Efficient Practices

- Minimizing overhead of managing revisions and change control

Benefits with implementing Agile

- Faster support within the team
- Better awareness of other's tasks
- Knowledge sharing on *howto* level
- Transparency around progress
- Visualization makes everyone aware of plan and status
- Better follow-up on commitments
- Better throughput due to work breakdown
- More realistic timelines
- More transparency around work overload
- Alignment in way of working with engineering teams if also management starts working in a new way



Read more on <http://www.stateofagile.com> and <http://www.succeedingwithagile.com/wp-content/uploads/2009/10/Reported-Benefits-of-Agile.ppt>

A person in winter gear, including a backpack and a walking stick, is seen from behind, climbing a steep, snowy mountain slope. The scene is misty and overcast, with the snow covering the entire landscape.

**"TO BE
OUTSTANDING,
GET
COMFORTABLE
WITH BEING
UNCOMFORTABLE."**

ALRIK KANDENBURG

Going truly Agile will make you feel uncomfortable at times.

And that is OK