

The image features a horizontal banner with a blue background on the left and a photograph on the right. The photograph shows a man in a black wetsuit holding a surfboard against a clear blue sky. The Protegra logo is overlaid on the blue background. The logo consists of the word "Protegra" in white, with the "g" in orange. Below it, the tagline "Business. Technology. Solutions." is written in a smaller white font, with "Solutions." in orange. The slogan "Be better" is positioned below the tagline in white.

Protegra
Business. Technology. Solutions.

Be better

Lean Project Change Management Tools and Techniques

Agenda

- **Agile/Lean Change Management Challenge**
- **Tools and Techniques**
- **Issue Management**
- **Recommendations**

Questions

- **How do we manage scope and change in an agile project?**
- **There is no question we will deliver more value, but how do we provide the visibility the clients need?**
- **A lot of the sources out there talk about looser project management with unconfirmed budget and scope. How can you ever get a project approved?**
 - Don't provide estimates until after 1-2 iterations are done
 - Execute the iterations until the budget is exhausted
 - Keep on adding scope as the project executes by trading other items

Protegra Givens:

- **We are a consulting company, some of the luxuries like not estimating the full project are not options for Protegra.**
 - We need to estimate the project for the clients
- **We need to inform the clients of changes. Sometimes Agile proponents advocate change, but we need a process to ensure the change is communicated and approved.**
 - Some times the change is recorded and implied that there are less formal processes
- **In short, we need to deliver what we estimated (promised) and use Lean techniques to deliver more.**
- **Scope is the currency for agile projects, budget is the currency for traditional projects. How do we modify of management processes?**

Waterfall



- **Provide an estimate for the entire project**
- **Take a significant amount of time and sign off detailed estimates**
- **Execute to that plan. Change request anything to that plan.**
- **Scope trading usually not an option. Client not usually given option to remove items as it complicates matters**

- **Objectives:**
 - Minimize surprises
 - Minimize changes
 - Maximize predictability

Lean or Agile



- **Provide an estimate for the entire project**
- **Take a material amount of time and produce solution and high level estimates**
- **Refines estimates as iterations proceed**
- **Execute to the iterations. Change request anything that cannot be accommodated.**

- **Objectives:**
 - Maximize Value
 - Encourage Change

Questions



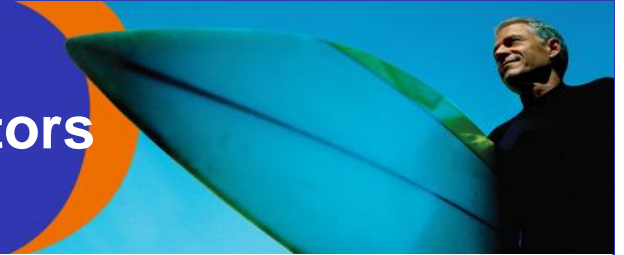
- **What is a CR and what is refinement? How should we account for this time?**
- **How do you balance Agile and change control?**
- **What is a scope enhancement? How can we make them defensible?**
- **Should we resolve ourselves to the reality and make a bucket of hours?**
- **What is an issue? How do you resolves issues and changes with agile practices.**
- **We want to be easy to work with and need to build trust! But sometimes we are don't CR everything we should.... And then that removes hours to work on legitimate refinements....**

Fundamental Issues



- 1. Trading Requirements and Refinement of Requirements can lead to more changes**
 - Hard to defend as to whether it is a change. Sometime you don't know all the assumptions you need to make and now you have more change!!!
- 2. Focus on Agility leads project to sometimes accept more change**
 - Want to encourage a win-win environment. Don't want to make it adversarial right away
 - We need to build up trust to fully do Agile. But this can lead to more accommodation and possibly profit loss.
 - Sometimes easier to just do it. Death by 1,000 cuts.
- 3. Agility sometimes leads clients to not give requirements the attention they deserve early on**
 - They know they can change it later if they don't get it right.
 - But this is anti-agile. This will cause waste and rework. We want the process to be able to adapt to required change later on, not change that wasn't required.

Management – Before, During, Leading Indicators



- **Before- Client Expectation Setting – Addresses #3**
 - Meet with Client and determine proper SDLC
 - 3 options - Lean, Iterative Development, Waterfall
 - Honest discussion of what value is expected
 - Lean isn't for all clients
- **During-There are two types of scope creep – Addresses #1, #3**
 - New or modified scope items that are easily defensible (NEW)
 - Two types:
 - Ones that are identified and documented by the team
 - Ones that are just absorbed by the team
 - New or modified scope items which are not easily defensible (MORE)
 - They are just MORE
 - More complex, More work, More involved
- **Leading Indicators- We need to track proactively – Addresses #2**
 - Burn down charts overall
 - EPI – Estimate Performance Index
 - Time Entry - ETC

During Best Practices



- **Scope Creep – New Guidelines**
 - All items added to or removed from the Product Backlog need to have a change request generated and approved. No exceptions.
 - User story name template – US ###.Date
 - All stories need to be estimated
 - Confirm all estimates with clients before start on Iteration
 - This gets buy-in on the level of estimates.
- **This is required moreso on an iterative development project. It should not be required on a true agile project.**

During Best Practices



- **Scope Creep – More Guidelines**

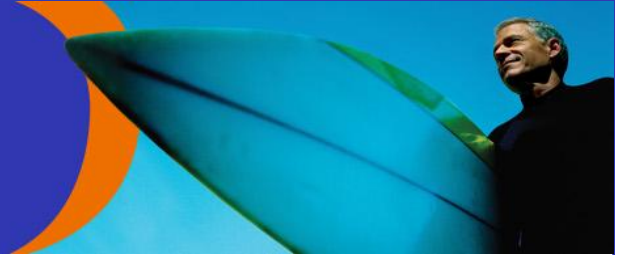
- Bucket that we propose and agree to manage jointly where estimates are not sufficient. (10-20% of project hours)
- Details
 - Hours at reduced rate!
 - Refundable if not used!
 - CRs will still occur for obvious scope changes
 - To be used for interpretation issues or more issues
 - Can be used to manage scope when level of complexity changes

Leading Indicators Guidelines

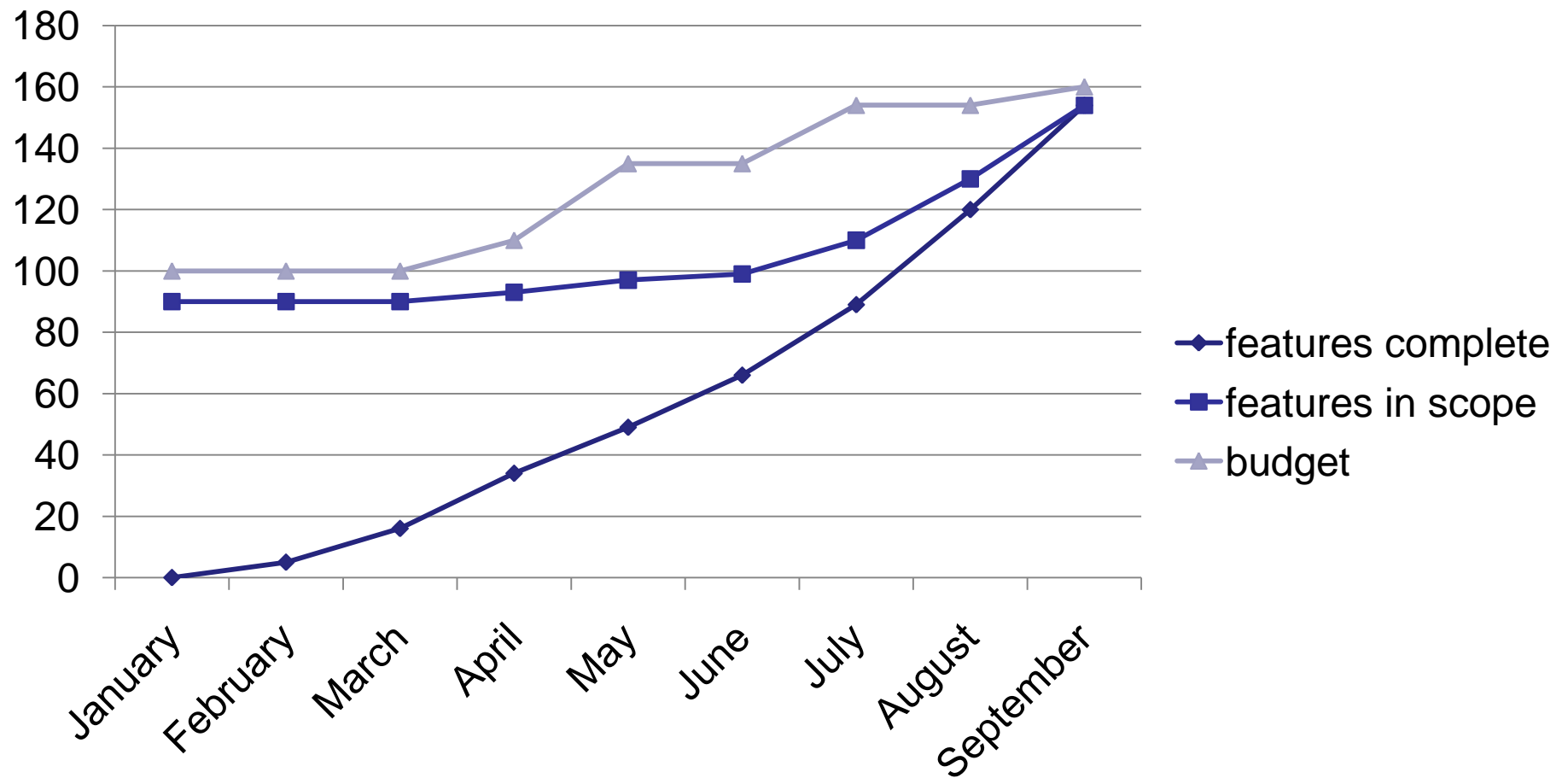


- **On the weekly status report, the following leading indicators will be reported:**
 - Estimate Performance Index = hours on schedule/hours actually spent
 - Overtime Performance Index = Overtime hours/normal hours
- **This is in addition to the weekly status report of budget, hours and general progress**
 - These items above are meant to track the specific type of scope baby crawl that occur in some agile projects
 - I call it baby crawl because it is cute at the start because it shows customer interaction and engagement. But unless you set up the gates by the stairs, scope can get away on you! 😊

Leading Graphical Indicator



- **Leading Graphical Indicator will be a burn up chart**



Burn up chart

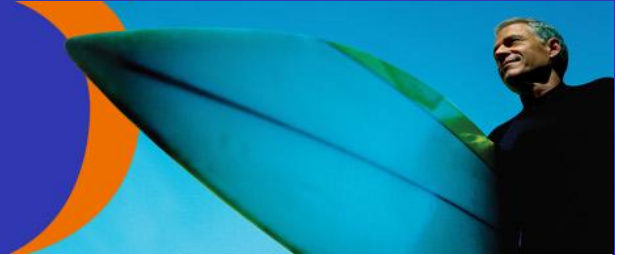
- Unlike a burn down chart which doesn't refer to budget, the burn up chart refers to budget and scope
- This gives constant feedback on the amount of work required to finish the project

Time Entry



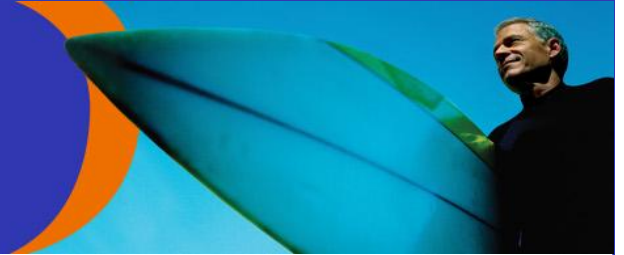
- **Time Entry – Estimate to complete**
- **All time entry submissions will require individuals to estimate their time to complete**
- **This will allow for the identification of issues of tasks that are not progressing**
- **Elimination of management buckets?**
 - Separate all Project Management, Application Architecture, Technical Architecture into tasks?
 - Maybe at least smaller tasks that provide a better window on progress
 - Sometime we lose progress tracking on large buckets like this.

Traditional Tools



- **Traditional Project Management Tools**
 - MS Project
 - WBS structure
 - Status Reports
 - Weekly Status Meetings

Project Management Tools



- **The Traditional Project Management tools are good at executing a plan that has limited change. They are not good for:**
 - Projects that have a lot of change
 - Projects that require constant re-planning
 - Project where everything is not known ahead of time
 - Projects where scope is the main currency for project execution, not budget.
 - Does not allow for the scheduling of defects. Too hard, too variable...
 - The project plan is disconnected from the rest of the team. It is seen as something just the PM handles.

Agile PM tools



- **Agile PM tools like Rally thankfully fit this gap**
 - Address the iteration planning activities and frequent changes
 - Integrates directly into the IDEs
 - Allows for assignment of both user stories AND defects
 - Create reports and visual project management that connects the whole team

Rally Demos



- www.rallydev.com

Issue Management



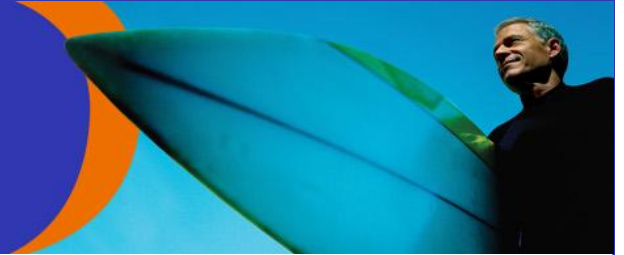
- **Just as important as change management is issue management.**
- **When do we escalate, when do we allow the team to resolve**
- **Like changes, we want to build trust and don't want to escalate too quickly or take too long.**

Issue Management Percussion



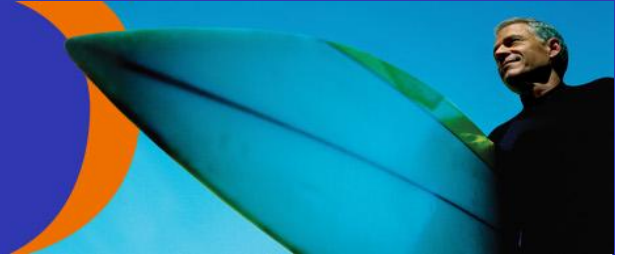
- **Issue:** An item related to the successful completion of the project that requires information, discussion, or a decision to resolve. The following activities will be done as part of the issue management percussion:
 - Project 'waits' are entered as issues if the team can not solve internally in 4 hours AND impact is either currently being experienced on the project or will be experienced on the project if not solved in 48 hours.
 - These issues will be categorized as yellow if some progress can still be made on this or other tasks. These issues will be categorized as red if progress is halted.
 - These issues will be entered with a due date
 - Unresolved Issues will be escalated to client PM at the end of each day. Date of escalation will be recorded.
 - Overdue issues will be escalated to project sponsor at the end of each day. Date of escalation will be recorded.
 - In addition issues will be reviewed on a weekly basis with the status report highlighting which issues are overdue.

Issue Management Percussion



- **Although we say that is the percussion, what happens if the team doesn't report issues?**
- **We may not be able to prevent it but how can we be aware of issues?**

Daily Stand up – Issue Management



- **Time Entry – Estimated Date of completion and blocking check box.**
- **All time entry submissions will require individuals to estimate their date of completion**
- **This will allow for the identification of issues of tasks that are not progressing and issues that have not been raised.**
 - Well it would take me 1 day, but I need my environment fixed.
 - I only have 2 days, but they have not signed off requirements yet