

## Sponsor/Product Owner Disengaged, Disempowered

- The right product owner must be Engaged, Knowledgeable and Empowered
- Educate product owner early on their role and the Agile process
- The right product owner has the most to gain or lose from project outcome

## Requirements too high level, too detailed or doesn't exist

- Educate team on different levels of requirements. Gather right level at right time
- Breakdown large epics into smaller valuable chunks
- Gather the 'details' just in time, avoid BRUF (Big Requirements Upfront)

## Team roles, skills, allocation not optimized for success

- Team roles must be able to get one feature/story all the way 'Done' (Feature Teams)
- Avoid or minimize multi-tasking and allocating people across several projects
- Acquire folks with the right skills. Educate them on being Generalizing Specialist

## Estimates not provided by actual team

- Involve actual team roles early during requirements gathering and estimation
- Estimates should use 'relative' sizing measures such as story points
- Project hours are calculated based on team size and estimated velocity per iteration

## Ineffective Release Planning and Iteration 0

- Don't skip these two critical steps! Don't spend 3 months on them either!
- Educate team early on effective release planning and iteration 0
- Engage the right system, DB, security, UI, architecture folks during iteration 0

## Testing and testers engaged later in the project, after development

- Testers are key to success and must be engaged early during release planning
- Acceptance tests are gathered as part of requirements not after development starts
- System and User Acceptance Testers work with team to test within each iteration

## Management being an impediment instead of removing them

- Management must focus on correcting team allocation issues, optimizing roles and improving team skills or training needed, providing proper tools and education
- Management should minimize team interruptions and remove daily impediments

## Project Manager focused on reporting 'green' to sponsor not on team

- Avoid creating 'light touch' project managers who are leading 10 projects
- Must learn about being ScrumMasters or identifying a good one for the team
- Create highly visible information radiators, immerse yourself with team

## Bad quality code with many defects emerges

- Educate team early on Agile/iterative development best practices such as TDD, automated builds, continuous integration, daily check-ins, design patterns
- Attack defects early on to reduce technical debt. Encourage design/code reviews

## Team not empowered, won't speak up or provide input

- Educate management and the team on the value of Servant Leadership
- Empower teams to be self organizing (not self leading!)
- Perform frequent retrospectives to get feedback, follow through on improvements

## No visibility into actual progress or daily issues

- Break work down into small iterations/cycles, measure progress iteratively
- Develop a clear definition of 'Done', measure and report what is 'done' visibly
- Have daily stand-ups to discuss completed/remaining work and impediments

## Lack of early architectural envisioning and proof of concept

- During iteration 0 engage the right folks to perform architectural envisioning
- Identify technical risks and create proof of concept stories or spikes to mitigate
- Develop high level process diagrams, UI Flows, data and architectural models

## Lack of iterative prioritization of Minimal Marketable Features

- Product owner must constantly be looking at the next most important MMF to produce
- Product owner must review work accomplished each iteration/cycle