

# Becoming an Agile Team Member



# About Me

- Doing agile since 1999
- Start ups / Enterprises
- Planigle - Consulting and Training
- Qcue – VP, Engineering
- Agile Austin - VP



# Becoming an Agile Team Member

- Often we focus on team practices
- Success of agile often comes down to individuals



# What's the Goal?

- To make meaningful, demonstrable progress on a daily basis so that we can learn quickly and take advantage of what we've learned





# Make It Visible

- Relate work to the goals
- Let others know what you're working on

Stories

Reports

Schedule

People

Changes

Refresh

Log out

Planigle

1.1

Iteration 4

Team A

All Owners

All Statuses

Number of Stories:

5

Velocity Allocation:

24 of 19.33 (124%) - Team A

Utilization:

▶ 23 of 19 (121%) - Team A

	Name	Owner	Size	Time	Status	Public	Rank	User Rank	
-	User searches for books by author, title or ISBN number	Sue Tester	8	7	In Progress	true	1	1	
	Search by title showing just titles	Bob Developer		2	In Progress				
	Add more details to results	Bob Developer		2	Not Started				
	Add search by author or ISBN	Bob Developer		1	Not Started				
	Test search	Sue Tester		2	Not Started				
+	User views detailed information on a book	Sue Tester	5	5	In Progress	true	2	4	
+	Administrator adds new books to site	Sue Tester	5	5	Not Started	true	6	5	
+	Administrator deletes book	Sue Tester	3	3	Not Started	true	7	6	
+	Administrator edits existing book info	Sue Tester	3	3	Not Started	true	8	7	

# Attack the Most Important Thing

- ... that you can help with.



# Break It into Small Chunks

- A day or less
- Enables feedback and collaboration



# Minimize Work in Progress

- Minimize the stories / tasks in progress
- Complete things before moving on





# Help Others

- Collaborate
- Share experiences
- Give feedback



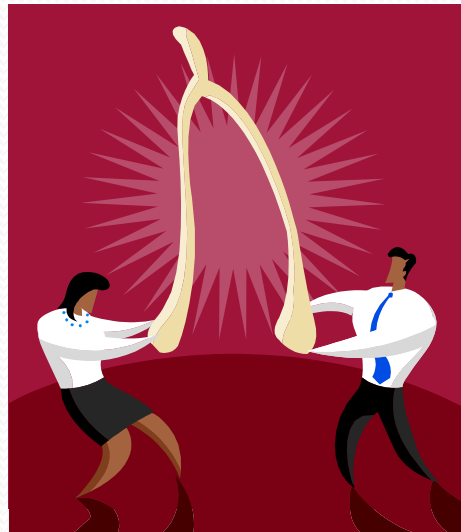
# Product Owner – Focus

- More detail on what is coming up
- Broader strokes on what is further out



# Product Owner – Small Chunks

- Constantly split off / eliminate less valuable items
- Makes it easier to prioritize and quickens feedback



# Product Owner – Quick Feedback

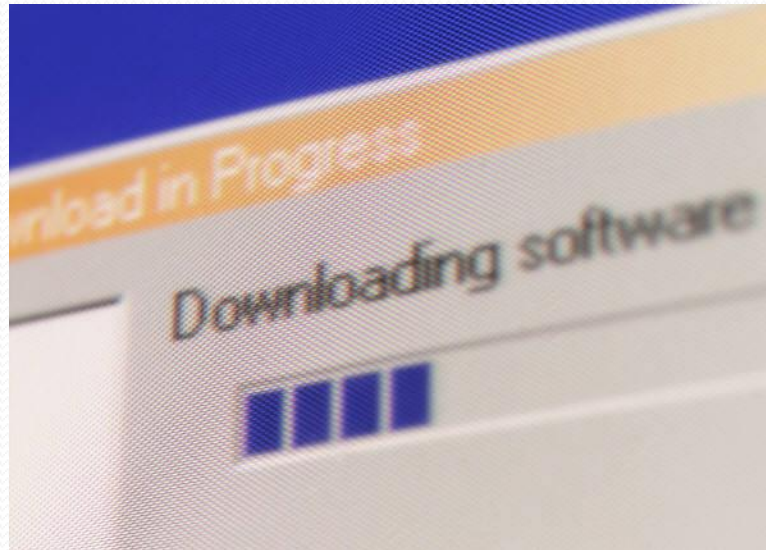
- Early and often
- Don't wait until the end





# Dev – Frequent Check-ins

- Check in at least once a day



# Dev – Shippable Code

- Check in unit tests with the code
- Each check in should be shippable



# Dev – Help Others

- Review code
- Share hints / experiences
- Don't be afraid to help on validation



# QA - Prioritize

- Focus on the most important issues





# QA – Quick Feedback

- Get involved early
- Don't be afraid of making multiple passes
- Give feedback early and often



# QA - Automate

- Make regression testing an easy / quick process



# Management - Facilitate

- No command and control
- Help the team help themselves



# Management – Process

- It is all about:
  - Opening communication
  - Focusing on the most important things
  - Getting better and better at it





# Management - Focus

- What is important / not important
- What is risky?
- Where are things held up?
- What prevents us from releasing?



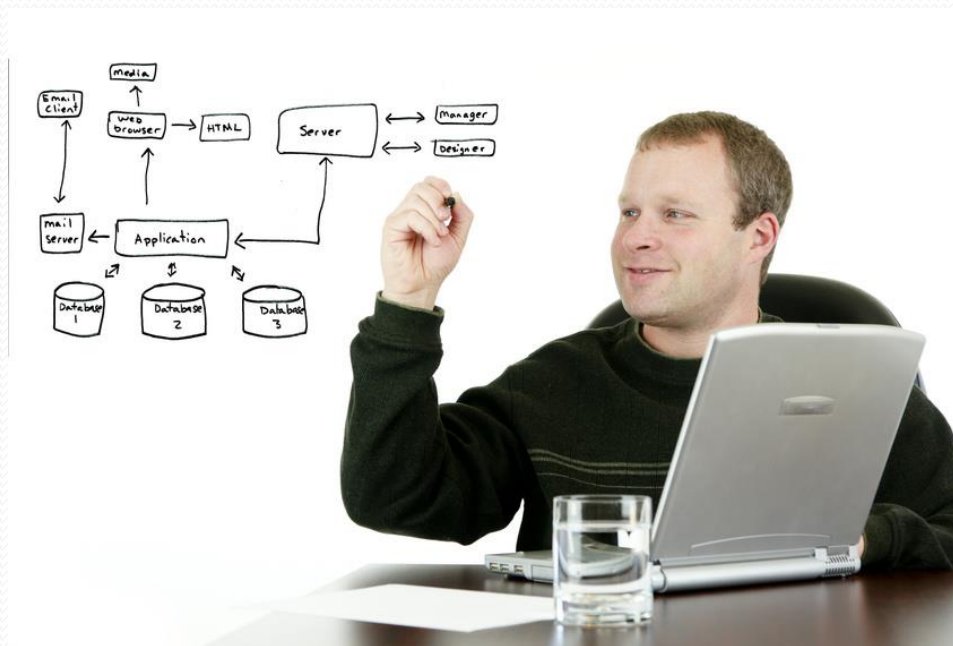
# Tech Writers

- Get involved early
- Don't be afraid of making multiple passes
- Give feedback early and often
- Keep up with the code



# Architects / Usability

- Focus on the biggest risks
- Provide runway for the team
- Involve the teams early and often



# Conclusion

- Make it visible
- Attack the most important thing
- Break it into small chunks
- Minimize work in progress
- Help others



# Resources

Walter Bodwell  
Planigle

[wbodwell@planigle.com](mailto:wbodwell@planigle.com)

Twitter: @wbodwell

[www.planigle.com](http://www.planigle.com)

[www.walterbodwell.com](http://www.walterbodwell.com)





# By Role Summaries



# Product Owner

- Make it visible
- Attack the most important thing
- Break it into small chunks
- Minimize work in progress
- Help others
- Focus – long vs. short term
- Give quick feedback

# Dev

- Make it visible
- Attack the most important thing
- Break it into small chunks
- Minimize work in progress
- Help others
- Frequent check ins
- Shippable code

# QA

- Make it visible
- Attack the most important thing
- Break it into small chunks
- Minimize work in progress
- Help others
- Give quick feedback
- Automate

# Management

- Make it visible
- Attack the most important thing
- Break it into small chunks
- Minimize work in progress
- Help others
- Facilitate, not command and control
- Process – how do we do things?
- Focus – how can we improve?

# Technical Writers

- Make it visible
- Attack the most important thing
- Break it into small chunks
- Minimize work in progress
- Help others
- Give feedback early and often
- Keep up with the code

# Architects / Usability

- Make it visible
- Attack the most important thing
- Break it into small chunks
- Minimize work in progress
- Help others
- Focus on the biggest risks
- Provide runway for the team
- Involve the teams early and often