



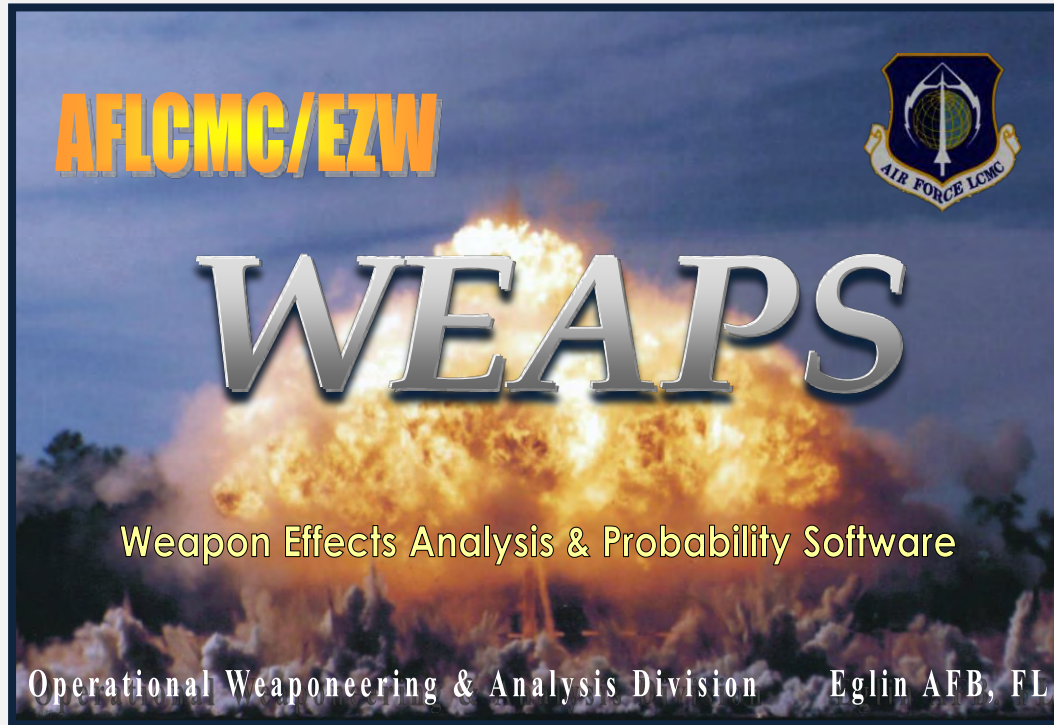
WEAPS

An Agile Story

Mr. Ash Stephens, SURVICE Engineering
AFLCMC-EZW



SURVICE
ENGINEERING COMPANY



A software system that establishes the effectiveness of inventory, developmental, and conceptual air-to-surface munitions.

Manifesto for Agile Software Development

We are uncovering better ways to develop software by doing it and helping others do it. Through this work we have come to value:

Responding to change <i>over</i> following a plan	Individuals and interactions <i>over</i> processes and tools
Working software <i>over</i> comprehensive documentation	Customer collaboration <i>over</i> following a plan

That is, while there is value in the items on the right, we value the items on the left more.

www.AgileManifesto.org

WEAPS – The Agile fix is in

Architecture – **No point without it**

- Maintainability and Scalability
- Use case development
- Decoupled trusted sub systems
- No more spaghetti code



Test Driven Development

- Acceptance Tests
- Unit Tests
- Automated nightly tests



Roles

- Product Owners
- Scrum Master
- Stakeholders
- Acceptance Testers
- Developers



Technology and Tools – **Latest versions**

- MS Team Foundation Server
- MS SQL Server
- Visual Studio



WEAPS – *WaterGiral*

- Developed then tested (Could go on forever)
- No source control repository
- Post processing and validation of output
- Took years for changes & release
- All changes had to be made before release
- Old technology delayed production data
- Hardware updated, software not
- Unhandled technical debt – spaghetti code
- Access DB used for tracking requirements
- Hard to get newer versions to test in-house
- ‘Workarounds’ & ‘tricking’ the application
- Zero Unit Tests



WEAPS – Agile

- Test Driven Development
- Industry standard change tracking tools
- Working software is the priority
- Frequent releases of the software to the field
- Priorities done first and released on schedule
- Newer versions have 60% time savings
- Uses the whole computer (multi-threading)
- Addresses technical debt often – re-architecting
- Web based tool for requirement tracking
- Automated builds & testing nightly
- Problems defined and solutions created
- ~ 900 Unit Tests with code coverage results



Agile – How it can help ACQUISITION

Agile Principles	⇒	<ul style="list-style-type: none">▪ All 12 don't have to be adopted at once▪ Add one or more at a time▪ Just don't stop doing one you have started
Contract Writing	⇒	<ul style="list-style-type: none">▪ Contract does not have to explicitly define process▪ Define problems and their respective priorities
Prioritize	⇒	<ul style="list-style-type: none">▪ Rack and stack priorities with stakeholder inputs▪ Most important requirements get worked first▪ Necessary evil – The hardest part, but necessary
Test Driven Development	⇒	<ul style="list-style-type: none">▪ Requirements – Define the problem▪ Acceptance Tests – Define the expected results▪ Let experts solve the problem by developing to a test
Team Dynamic	⇒	<ul style="list-style-type: none">▪ Define it and understand it▪ They are all different

Agile – The Push Back...And a response

“I can’t.”

Finish the sentence... “I can’t YET”.

“We aren’t there yet.”

No, you aren’t there yet and that’s ok. Just do one thing first.

“Chasing water falls.”

Agile does not chase anything. It responds to change, unlike “*WaterGiral*”.

“I don’t know about all that.”

Listen to experience! Doing it, and messing up makes you better...Not Googling it.

“It’s just another system development life cycle process idea.”

Agile IS NOT a process. It’s a manifesto with principles to form your own.

Conclusion

The Agile Manifesto and its principles were made for software.

But...