

Agile and Software Quality Assurance

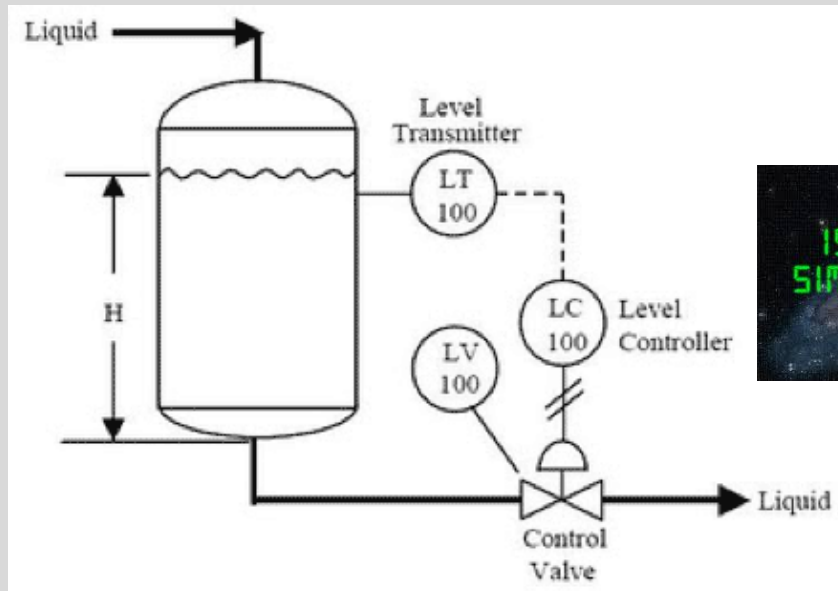
Challenging Perceptions

Lance Abbott
Chief Technical Advisor
Savannah River Nuclear Solutions
Savannah River Site

Allow me to introduce myself...



NQA-SQA Subcommittee





OPINION

**The information contained in this presentation is the opinion of the presenter.
It is not necessarily representative of my employer or the NQA-1 software
subcommittee**

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

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



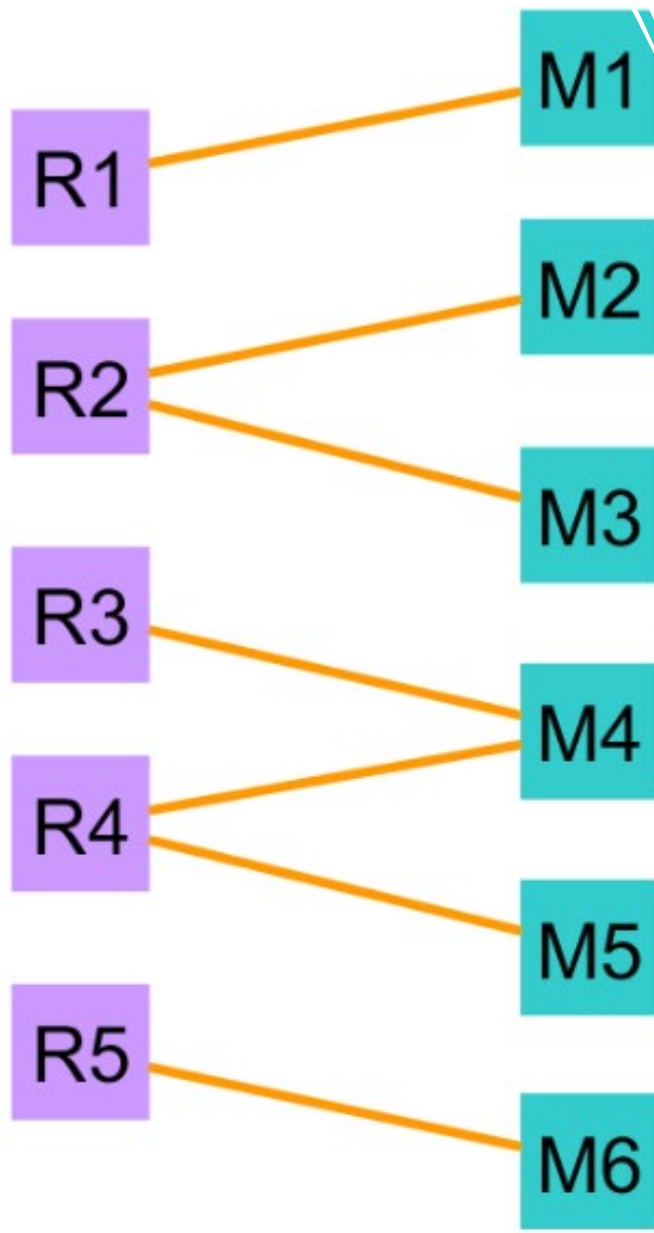
Agile Principles

- **Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.**
- **Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.**
- **Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.**
- **Businesspeople and developers must work together daily throughout the project.**
- **Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.**
- **The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.**



Agile Principles

- **Working software is the primary measure of progress.**
- **Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.**
- **Continuous attention to technical excellence and good design enhances agility.**
- **Simplicity--the art of maximizing the amount of work not done--is essential.**
- **The best architectures, requirements, and designs emerge from self-organizing teams.**
- **At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly**



**Will you just map
AGILE to NQA-1 for
me?**



Agile Methodologies

Scrum

Kanban

Scrumban

DevOps

(Bus)DevOps

Design Thinking

Agile Project Management
(AgilePM)

PRINCE2 Agile

PMI-Agile Certified Professional
(PMI-ACP)

Project Half Double

Agile Program Management
(AgilePgM)

Scale Agile Framework (SAFe)

Large-Scale Scrum (LeSS)

Nexus

Scrum at Scale (S@S)

Spotify Model

Scaled Agile Lean Development
(ScALeD)

AgileSHIFT

Agile Fluency

Open Space Agility (OSA)

Agility Scales

Holocracy

Sociocracy

Disciplined Agile (DA)

Praxis

Toyota Production System (TPS)

Agile Digital Services (AgileDS)

Management of Portfolios (MoP)

Standard for Portfolio
Management (SfPfM)

Agile Portfolio Management
(AgilePfM)

Evidence-Based Portfolio
Management (E-B PfM)

Bimodal Portfolio Management
(Bimodal PfM)

eXtreme Programming (XP)

Acceptance Test Driven
Development (ATDD)

Test-driven development (TDD)

Behavior-Driven Development
(BDD)

Feature-Driven Development
(FDD)

Experiment-Driven Development
(EDD)

User Experience Design (UX
Design)

Agile Business Analysis (AgileBA)

Continuous
Integration/Continuous
Deployment (CI/CD)

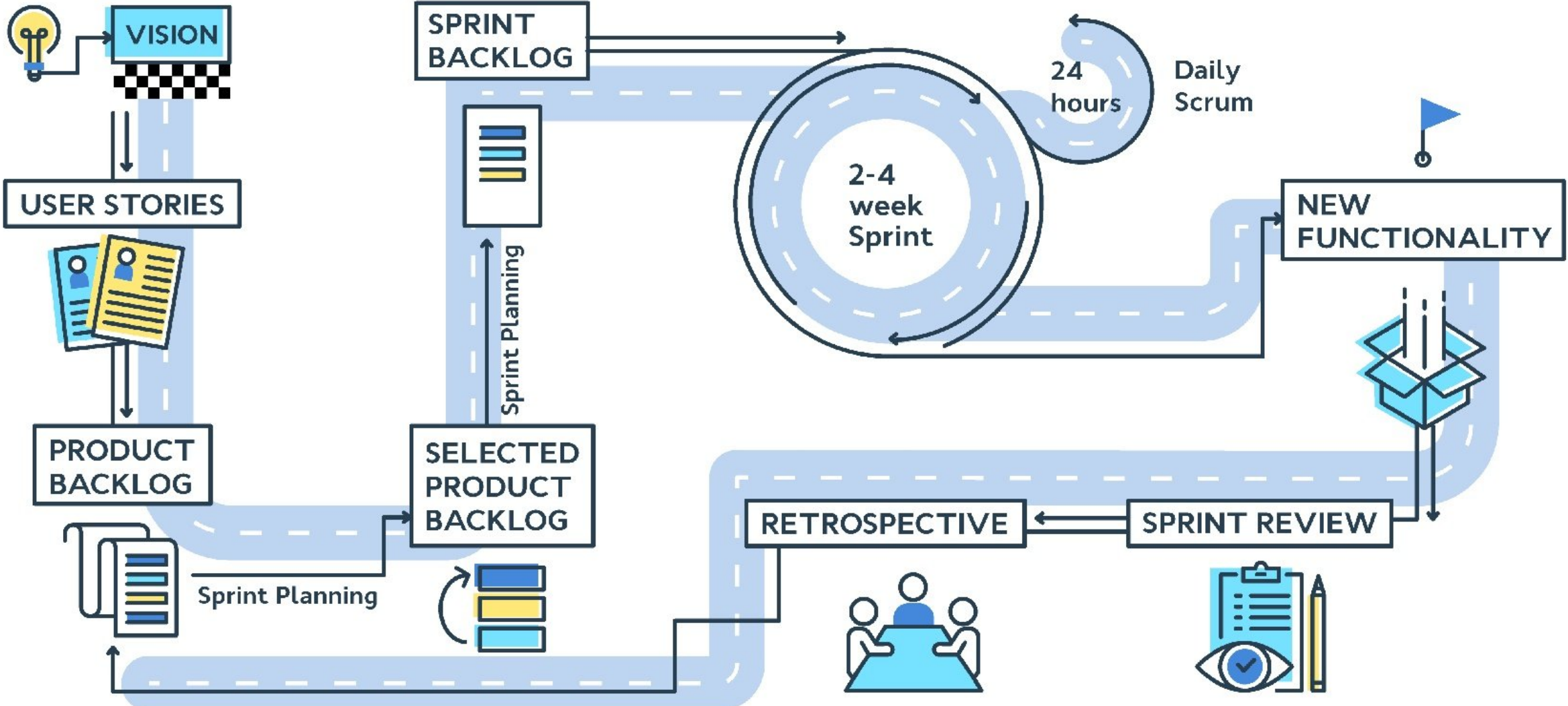
Agile Modeling (AM)

OK, Scrum is probably the most popular, how about that?

9



SCRUM PROCESS





Are “AGILE” and
NQA-1 / DOE 414.1D
Compatible?



HOW DO WE
DO IT ?

At SRNS, we include the quality deliverables as tasks in our Sprints, like all other work



In my experience

**Developers
believe...**


Developers want to produce quality, valued software

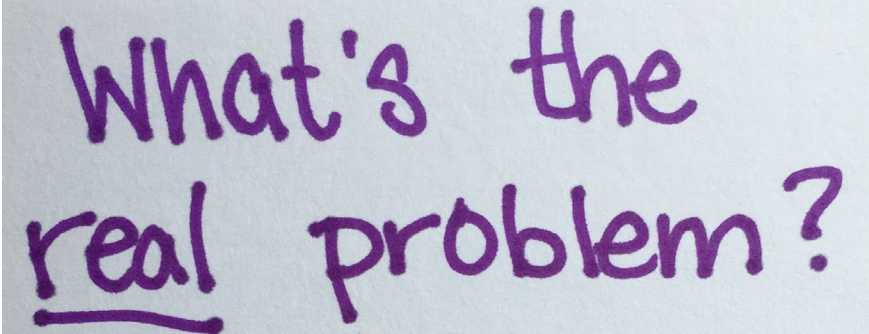
Quality processes today are built into the toolsets they use.

SQA DOCUMENTS are largely a duplication of work that do not provide value

Getting data out of the tools to send to “Records” is painful, time consuming, and provides little value

We produce SQA documents because we are required to, not because we believe that they improve the product

- 
- **We (SQA Professionals) don't know how to audit modern software development methodologies?**
 - **People use the word "Agile" as an excuse avoid work that they don't want to do?**
 - ***SOME* SQA personnel are out of touch with how modern software is developed and how software development is taught?**
 - **We've always audited documents, so it must be the only acceptable way?**
 - **From a developer's point of view - Duplication of Efforts / Wastefulness**



What's the
real problem?

Under what environment were the current SQA requirements created?

- Documents were handwritten
- Given to a typist to prepare
- Routed for comments
- Comments handwritten on document, usually one reviewer at a time
- Incorporated
- Routed for “wet” signatures





**“What if we don’t change anything at all ...
and something magical just happens.”**



The Magic!

- **Agile Guidance for Software Quality Assurance?**
- **Better understanding of how Software Development is being done and taught and how that should be reflected in standards, orders, and audits**
- **Removal of the document – centric approach**



time for

discussion