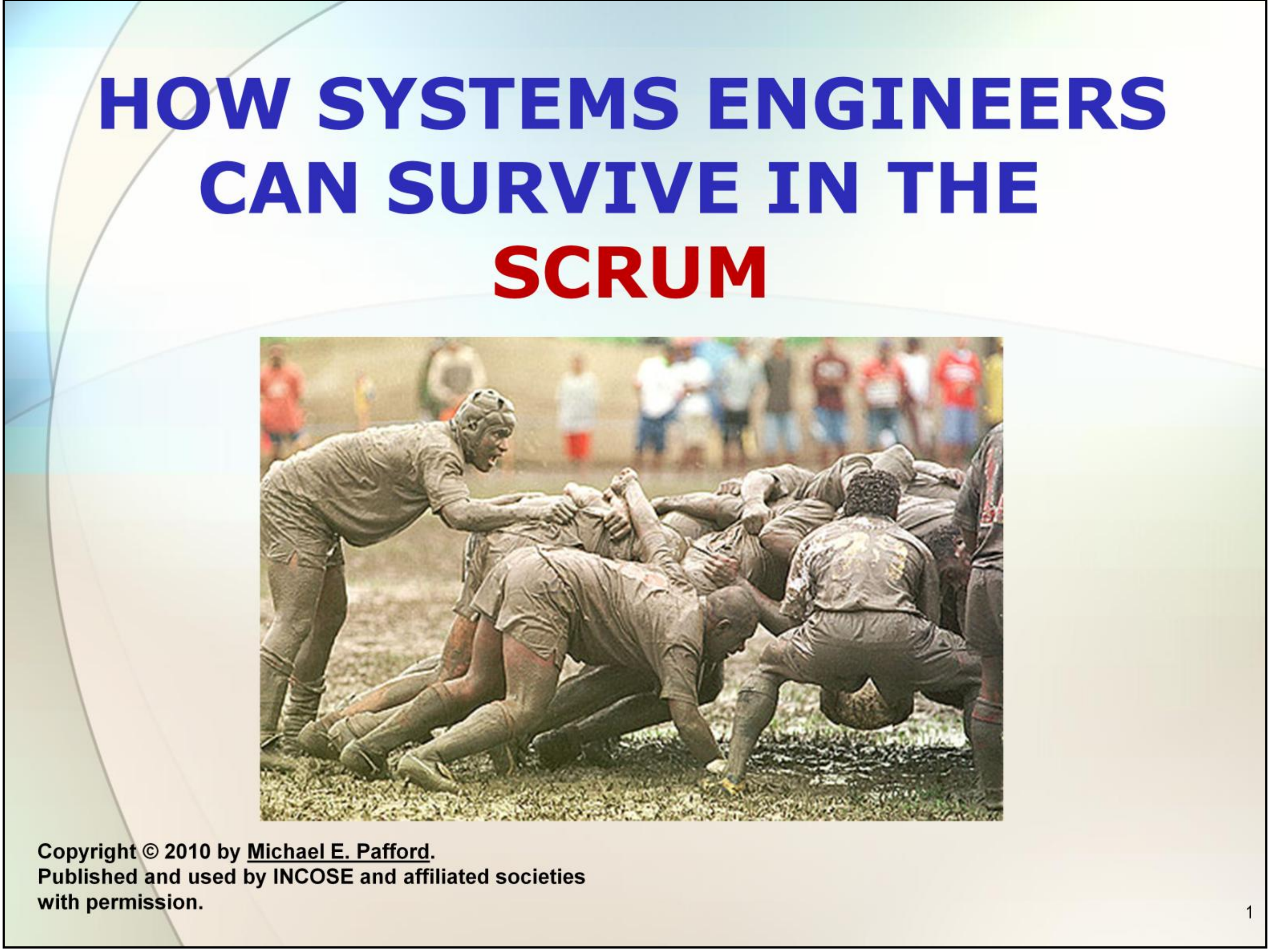
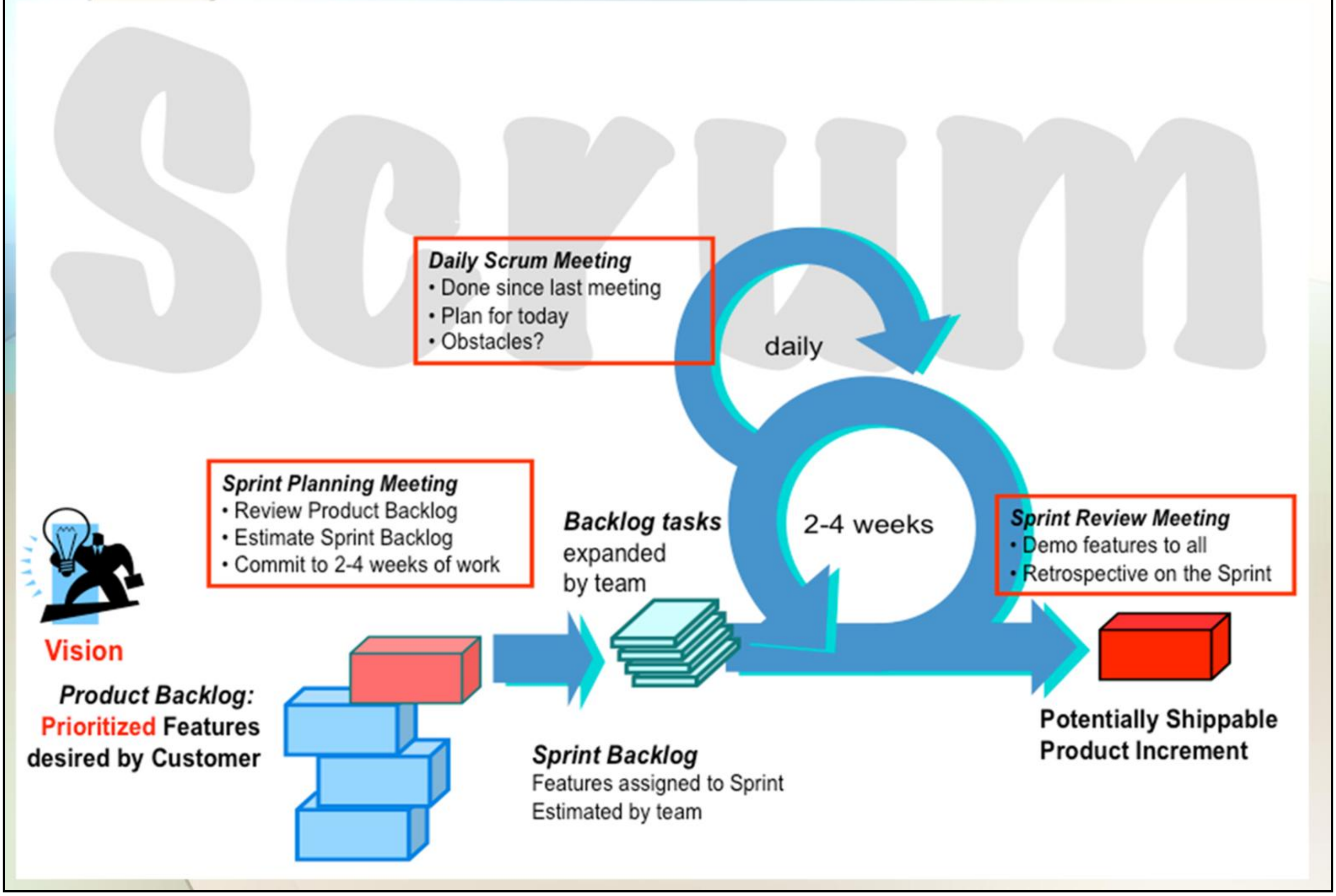


HOW SYSTEMS ENGINEERS CAN SURVIVE IN THE SCRUM

In this presentation I’m going to pass along some of the many Lessons Learned I collected while serving as a ‘traditional’ Systems Engineer on a Project Team developing a Software-Intensive System using the agile software development process called SCRUM.

WHAT IS SCRUM, AND WHERE DID IT COME FROM?



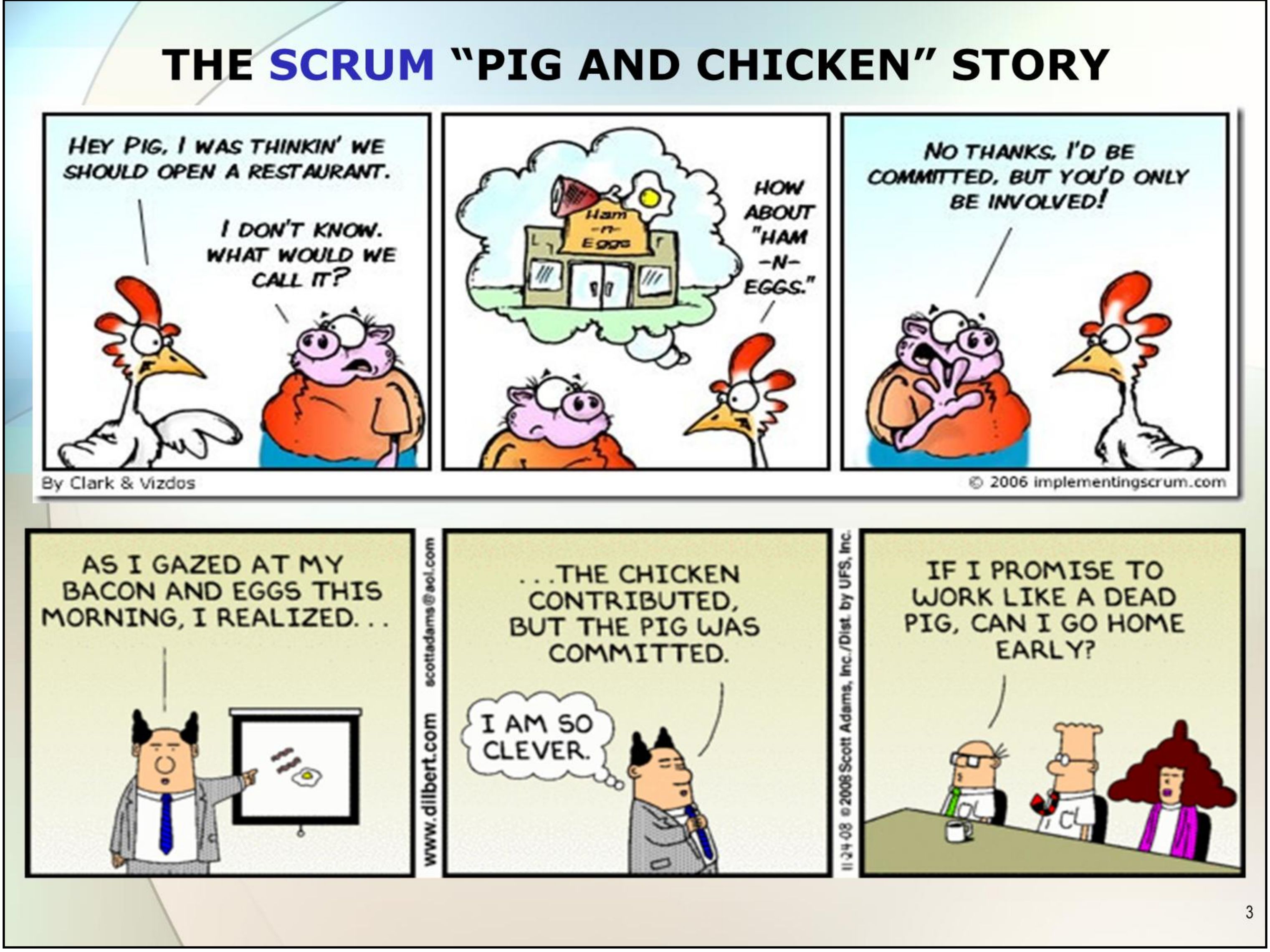
Scrum is an iterative, incremental process for developing any product or managing any work.
It produces a potentially shippable set of functionality at the end of every iteration.

It's attributes are:

•an agile SW development process to manage and control development work.
•a team-based approach to iteratively, incrementally develop systems and products when requirements are rapidly changing
•scalable from single projects to entire organizations.
•has controlled and organized development and implementation for multiple interrelated products and projects.

The short history of SCRUM methodology with the most important dates:

•In 1986 Hirotaka Takeuchi and Ikujiro Nonaka described a new approach to product development in which all the phases of the process overlap and the team works together across the different phases.
•In 1991 DeGrace and Stahl in “Wicked Problems , Righteous Solutions” referred to this approach as Scrum (from the rugby term scrummage).
•In the early 1990s the Scrum methodology was introduced to companies by Ken Schwaber and it was for the first time called SCRUM by Jeff Sutherland, John Scumniotales, and Jeff McKenna.
•In 1995 Jeff Sutherland and Ken Schwaber described Scrum and presented the results to the Object Management Group (OMG) at a yearly conference devoted to object-oriented programming.
•2001 “Agile Software development with Scrum” book written by Ken Schwaber and Mike Beedle



THE SCRUM "PIG AND CHICKEN" STORY

HEY PIG, I WAS THINKIN' WE SHOULD OPEN A RESTAURANT.
I DON'T KNOW. WHAT WOULD WE CALL IT?
Ham -n- Eggs
HOW ABOUT "HAM -N- EGGS."
NO THANKS, I'D BE COMMITTED, BUT YOU'D ONLY BE INVOLVED!

By Clark & Vizdos

© 2006 implementingscrum.com

AS I GAZED AT MY BACON AND EGGS THIS MORNING, I REALIZED...
...THE CHICKEN CONTRIBUTED, BUT THE PIG WAS COMMITTED.
I AM SO CLEVER.
IF I PROMISE TO WORK LIKE A DEAD PIG, CAN I GO HOME EARLY?

www.dilbert.com scottadams@aol.com

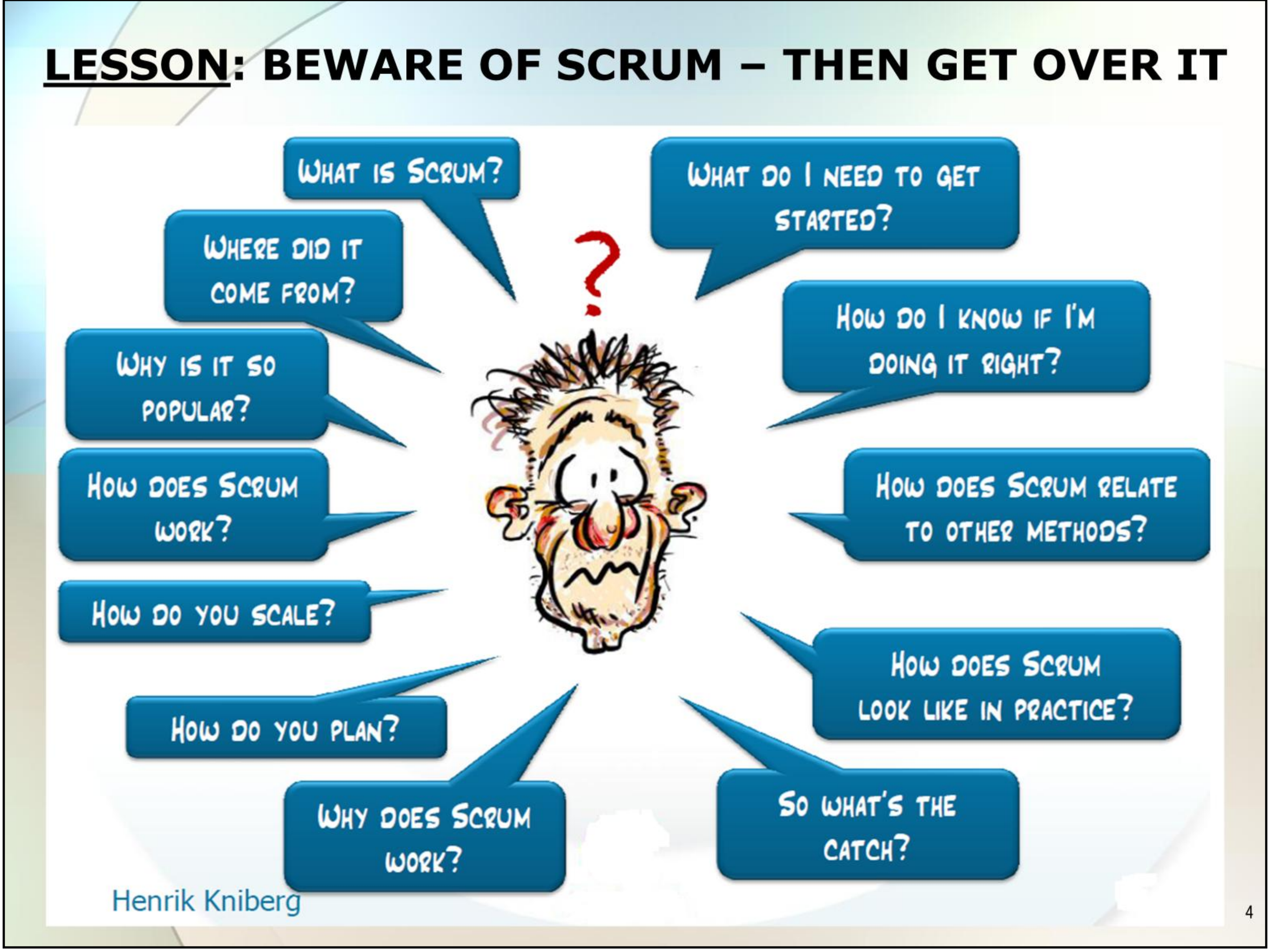
11-24-08 © 2008 Scott Adams, Inc./Dist. by UFS, Inc.

In his book, "Leadershift: Reinventing leadership for the age of mass collaboration", Emmanuel Gobbillot identifies two types of roles in the Scrum process—pigs and chickens. I'll of course talk about the 'real' Scrum roles in a minute. Meanwhile, the story. I'm sure most if not all of you know it. It's shown in the top cartoon and lampooned by the one and only Dilbert in the bottom one.

The joke does have a message about the nature of roles in projects involving Scrum.

The pigs are the ones committed to the project, with `their bacon on the line,' while the chickens are involved because they are interested in its benefits.
The pigs are running the scrum. They are the builders and doers.
The chickens provide impetus through their desires and needs but it is not in their interest to get in the way of the process.
The chickens are usually expected to provide—dare I say it—their 'eggspertise' when needed.

Systems Engineers start out as Chickens for the Scrum Restaurant/Breakfast.
I think you'll understand after this talk, how SEs wind up constantly bouncing between being the involved Chicken (QA, CM, etc.) and committed Pig (requirements engineering, testing, etc.) during the lifecycle of the project.



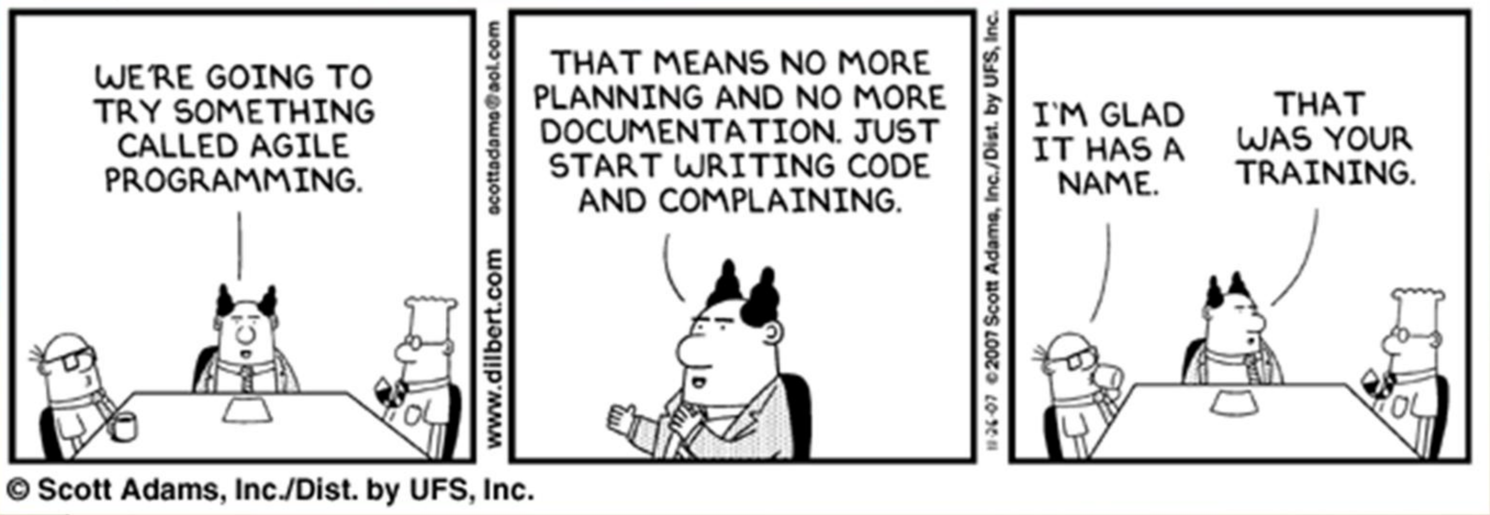
LESSON: BEWARE OF SCRUM – THEN GET OVER IT

Henrik Kniberg

Implementing SCRUM includes:

- Getting started immediately–and “getting good” fast
- Overcoming individual resistance to the changes Scrum requires
- Staffing Scrum projects and building effective teams
- Leading self-organizing teams
- Making the most of Scrum sprints, planning, and quality techniques
- Understanding Scrum’s impact on HR, facilities, systems engineering, and project management

SCRUM CAN SCARE SYSTEMS ENGINEERS


WE'RE GOING TO TRY SOMETHING CALLED AGILE PROGRAMMING.
THAT MEANS NO MORE PLANNING AND NO MORE DOCUMENTATION. JUST START WRITING CODE AND COMPLAINING.
I'M GLAD IT HAS A NAME.
THAT WAS YOUR TRAINING.
www.dilbert.com
scottadams@aol.com
11-26-07 © 2007 Scott Adams, Inc./Dist. by UFS, Inc.

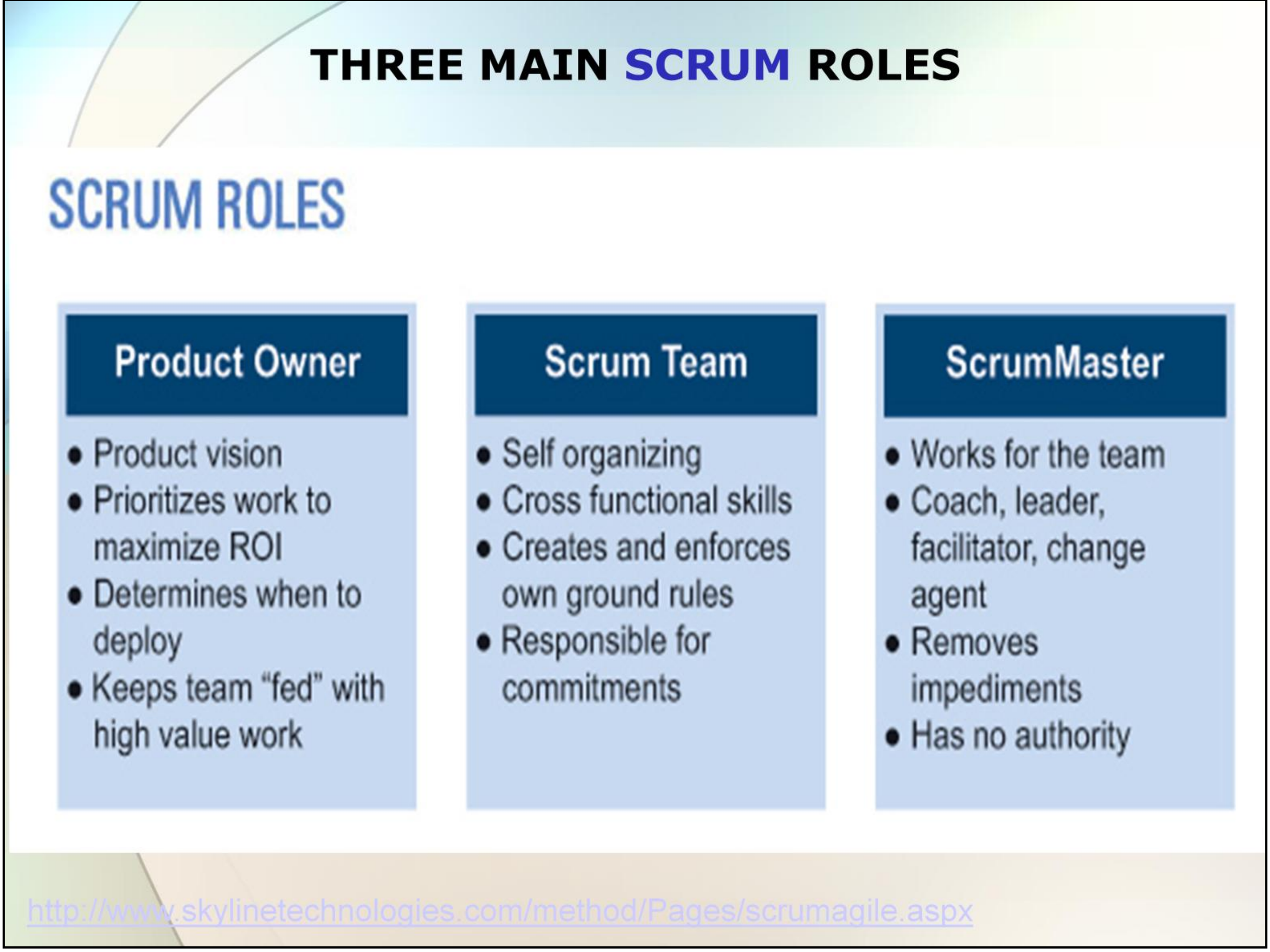
© Scott Adams, Inc./Dist. by UFS, Inc.

To a traditional disciplined systems engineer, this cartoon is like a nightmare!

The first time I heard this kind of planning for the project we'd be working on, I immediately thought to myself, "This is going to fail outright, or I really am going to die trying to maintain the kind of project engineering rigor I'm pretty sure our customers are expecting".

As usual, our customers were to a person DoD and Government Test and Evaluation technicians, engineers, and managers; every one having 'grown up' in their respective areas as educated and/or trained Systems Engineers. Very few of our customers had any strong SWE or DBE backgrounds—much less Any knowledge of agile/Scrum methodologies. And in the beginning of the project I was the primary customer liaison, leading the collection, refinement, and translation of needs and requirements.

As it turned out, another professional Scrum Team successfully 'educated' Program/Project Management, Systems Engineers, and Customer stakeholders, 'carried' us involved chicken Systems Engineers through the project, and delivered a working prototype of the required system that satisfied the customers.



THREE MAIN SCRUM ROLES

SCRUM ROLES

Product Owner

- Product vision
- Prioritizes work to maximize ROI
- Determines when to deploy
- Keeps team “fed” with high value work

Scrum Team

- Self organizing
- Cross functional skills
- Creates and enforces own ground rules
- Responsible for commitments

ScrumMaster

- Works for the team
- Coach, leader, facilitator, change agent
- Removes impediments
- Has no authority

http://www.skylinetechnologies.com/method/Pages/scrumagile.aspx

This slide shows the three main Roles for small to medium sized projects toward software intensive systems that employ agile/Scrum as a methodology.

I’ve given the reference for this graphic. But, you’ll find that most of the Scrum literature bins the roles basically the same way.

I’ll go into a little more detail about each role in the next few slides.

It’s probably good here to sort of lay out which project stakeholders played which Scrum Roles in the project I have the most recent experience in:

Product Owner(s)	Internal Project/Program Manager and External DoD/Government Customers
Scrum Team	Pigs: SWEs and DBEs Chickens: SEs, QA, CM, and Documentation Experts
ScrumMaster(s)	SWEs and DBEs—they were the uberPigs!

LESSON: YOU TOO CAN BE A SCRUM PRODUCT OWNER

The Scrum Product Owner is responsible for the ROI of the project.

This is the person representing everyone’s interest in the project and is responsible for the project delivering expected value.

The Product Owner is usually the Customer who commissions the project and owns the deliverables.
The Product Owner’s role is to prioritize the functional and non-functional requirements for the project.
The Product Owner aggregates input from users, stakeholders, and other interested parties to form a single view of the prioritized requirements for the system.

The right systems engineer may be a very good candidate for a Scrum Product Owner role.

In many organizations, systems engineering is viewed as the customer for software engineering.

Ken Schwaber in his book, Agile Project Management with Scrum, describes the relationship of the Scrum team and the Product Owner as, “constantly collaborating, scheming together about how to get the most value for the business”.

This is the model of how systems engineering, software engineering, and support organizations should be operating for effective agile/Scrum SW development – working together with a common goal of value for the organization.

Schwaber, Ken. Agile Project Management with Scrum . Microsoft Press, 2004.

LESSON: YOU TOO CANNOT BE A SCRUM MASTER

The ScrumMaster is usually not the Systems Engineer, Project Manager, or Program Manager. It's usually a senior SWE or DBE. The ScrumMaster is a **Process Manager**, responsible for ensuring that the Scrum SW process is used as intended.

He/She acts more like a Coach, responsible for cheering the Scrum Team on, for being the Team Leader, and the Team Guide.

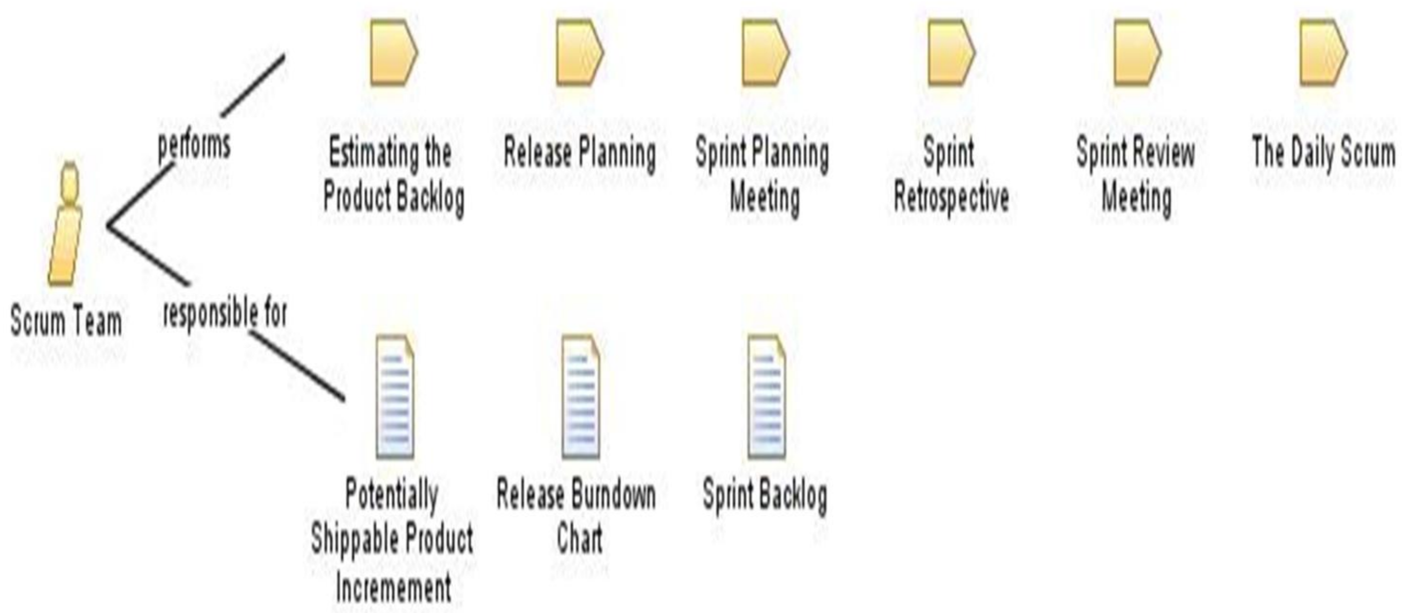
The ScrumMaster is also like a Referee, ensuring all Scrum Team members follow the rules of the process.

It's the job of the ScrumMaster to remove any impediment, within or external to the Team, that prevents team members from reaching their goal of building the software they commit to at the beginning of each Scrum Sprint cycle.

For the life of the project the ScrumMaster is a Facilitator:

- Improving the lives and productivity of the development team by facilitating creativity and empowerment.
- Enabling close cooperation across all roles and functions and removing barriers.
- Shielding the team from external interferences and removing "Impediments".
- Ensuring that the process if followed.
- Inviting appropriate people to the daily Scrum meetings, iteration reviews, and planning meetings.
- Removing the barriers between development and the Customer so that the Customer directly drives the functionality developed.
- Teaching the Customer how to maximize ROI and meet objectives through Scrum.
- Improving the SWE practices and tools so each increment if potentially shippable.

LESSON: YOU TOO WILL BE A SCRUM TEAM MEMBER



The Scrum Project Team, or Project Team, or just The Team, is a cross-functional group of people with all the different skills needed to turn requirements into something that is an increment of potentially shippable functionality.

The team includes analysts, designers, architects, coders, database experts, QA, CM, testers, etc.—and most of these are roles that Systems Engineers routinely perform.

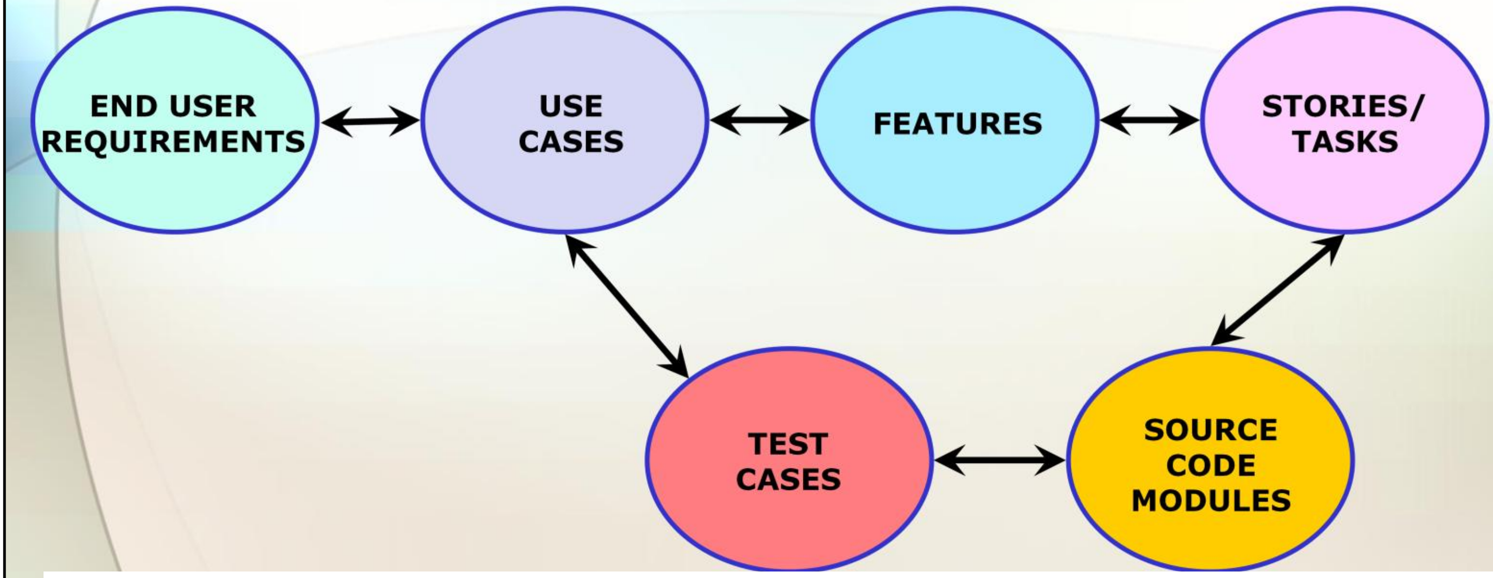
The Team actually commits to the Scrum Product Owner what they' will do every iteration. At the end of each iteration they show results to the Product Owner and then the Product Owner can decide what to do next.

Effective Scrum Teams are usually Seven, plus or minus two members that organize themselves and their work—with the ScrumMaster facilitating. When their work is done they demonstrate it to the Product Owner.

Systems engineering is still required with agile development.
Agile/Scrum certainly doesn't require fewer written requirements—which means, SE expertise needed.
As collaborative as Scrum is, there are still some very important sequential activities that must happen. For example, systems engineering must do a high-level first pass of requirements and allocate them to major incremental releases before the Scrum Team SW and Database developers get going on their Stories and Tasks.

LESSON: POLISH UP ON REQUIREMENTS ENGINEERING

NOTIONAL “TRACE DIAGRAM”



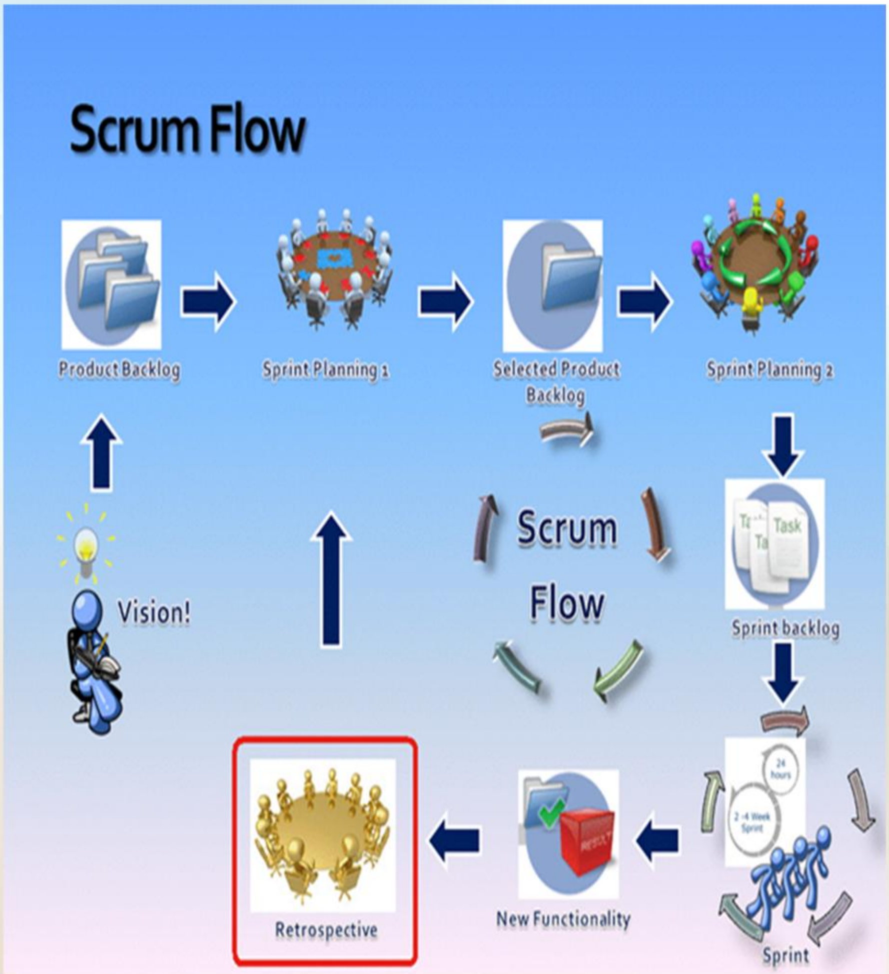
- This slide shows what turned out to be my (the System Engineer’s) most important value added to the Scrum Project—maintaining traceability of all project ‘artifacts’ throughout the project lifecycle.
- This is the strongest example I can give you, of when the involved SE Chicken becomes the committed SE pig.

**** Very Important: You’re no longer turning Needs into Requirements (or Capabilities in the case of JCIDS) and Specifications , then maintaining that traceability. In Scrum you’ll be helping to transform Needs to Requirements, then into Functions, Features, and Stories/Tasks, which themselves become SW modules you’ll help test at the Feature, Component, and System levels. ****

- The SE is primarily responsible for gathering, analyzing, and refining the initial set of Customer or End User Requirements. Systems Engineers with Operational Experience continue to be ideally suited for this role.
- The Scrum Team also relies on the SE to lead the next step; formulating Use Cases for the proposed system. This is the first opportunity for the SEs, SWEs, and DBEs to design the system functionality in a ‘language’ (i.e., Use Case notation) that is somewhat common to all. ** Keep in mind that some people like narrative Use Cases, other prefer Graphical. **
- The SE/SWE/DBE Team continues to collaborate on turning the Use Case Functions into System Features; narrative lists of things the Developers know that SW should be able to do.
- The SE/SWE/DBE Team then (emphasis is starting to shift now from SE to SWE and DBE) creates from the Features a set of Stories (Feature Groups) and Tasks (specific Features that can be used in one or more Feature Groups). It is these Stories and Tasks that Team members will take ownership to do each day.
- Each Story or Task becomes a SW module, created by the SWEs and DBEs. At this point the SE is staying in close contact to maintain the trace Back from the code being written to the Stories and Tasks. They’re already maintaining the other traces back to Requirements.
- The SE becomes more involved when the developers have written code, tested it at the module level, and are ready to test at the Features and System level against the Use Cases.

Keeping up with this Traceability (not easy by any means) will Guarantee your (SE) continued close

LESSON: IT’S CALLED “SPRINT” FOR A REASON



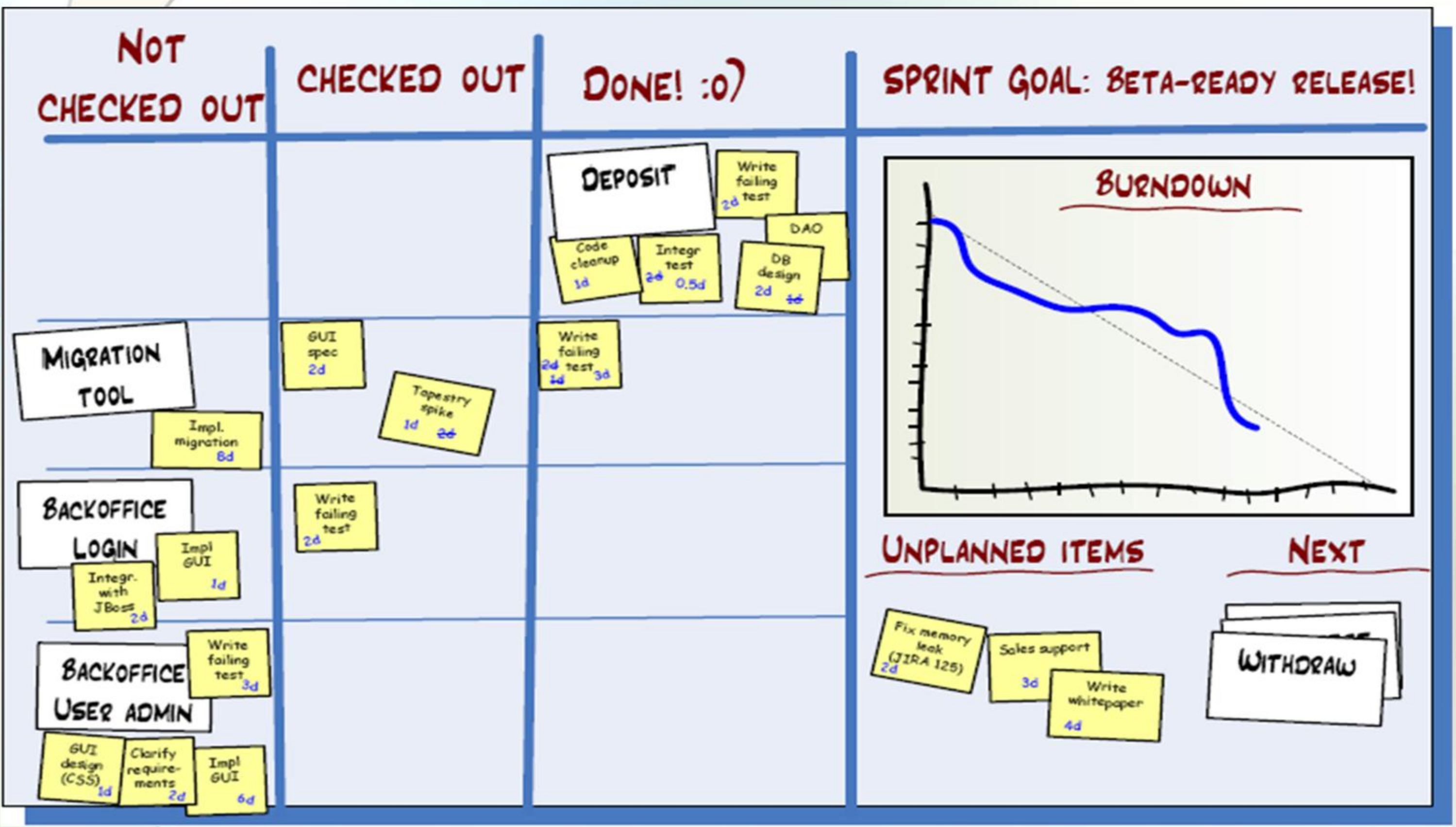
Sub-Lesson: You can have SE Scrum Sprints, Parallel SWE/DBE and SE Scrum Sprints, and Combined SE and SWE/DBE Scrum Sprints.

- On the left is a Rugby player. He‘s just gotten the ball kicked to him from out from under a Scrummage. He’s Sprinting towards the end line to attempt a Try (spiking the ball or his upper body in the scoring zone for a Try (5 points)). Its never easy.
- On the right is another depiction of the Scrum Flow, including Sprint Planning, Sprint Backlog Development, and Sprint [SW development] against the Backlog.

All of this happens under strict daily and weekly Sprint time constraints.

- **A Very Strong Lesson Learned during this project**: In order for an SE to ‘keep up’ with the rest of the Scrum Team, you have to be ready to 1) couch all SE tasks in the same Stories/Tasks form the other Scrum Team members use and 2) accomplish those tasks in the same constrained timeline as the rest of the team. Scrum waits for no man-or woman.
- In the beginning of the Project I’m referring to, the first 2-4 week Sprint was designated an SE Sprint. The reason being it was most important for the Needs and Requirements to be captured and refined. So, the team developed Stories and Tasks along those lines. SWE and DBE team members hung back for the most part until the SE Sprint was completed.
- During the SE Sprint we still held 15-minute Sprint Meetings each where each of us SEs answered the same three basic questions: What have I done since the last meeting?, What do I plan to do today?, and Do I have any Impediments?
- During the project we had SE, Parallel SE and SWE/DBE, and Combined Sprints.

LESSON: TRY TO STAY ON THE SCRUM BOARD



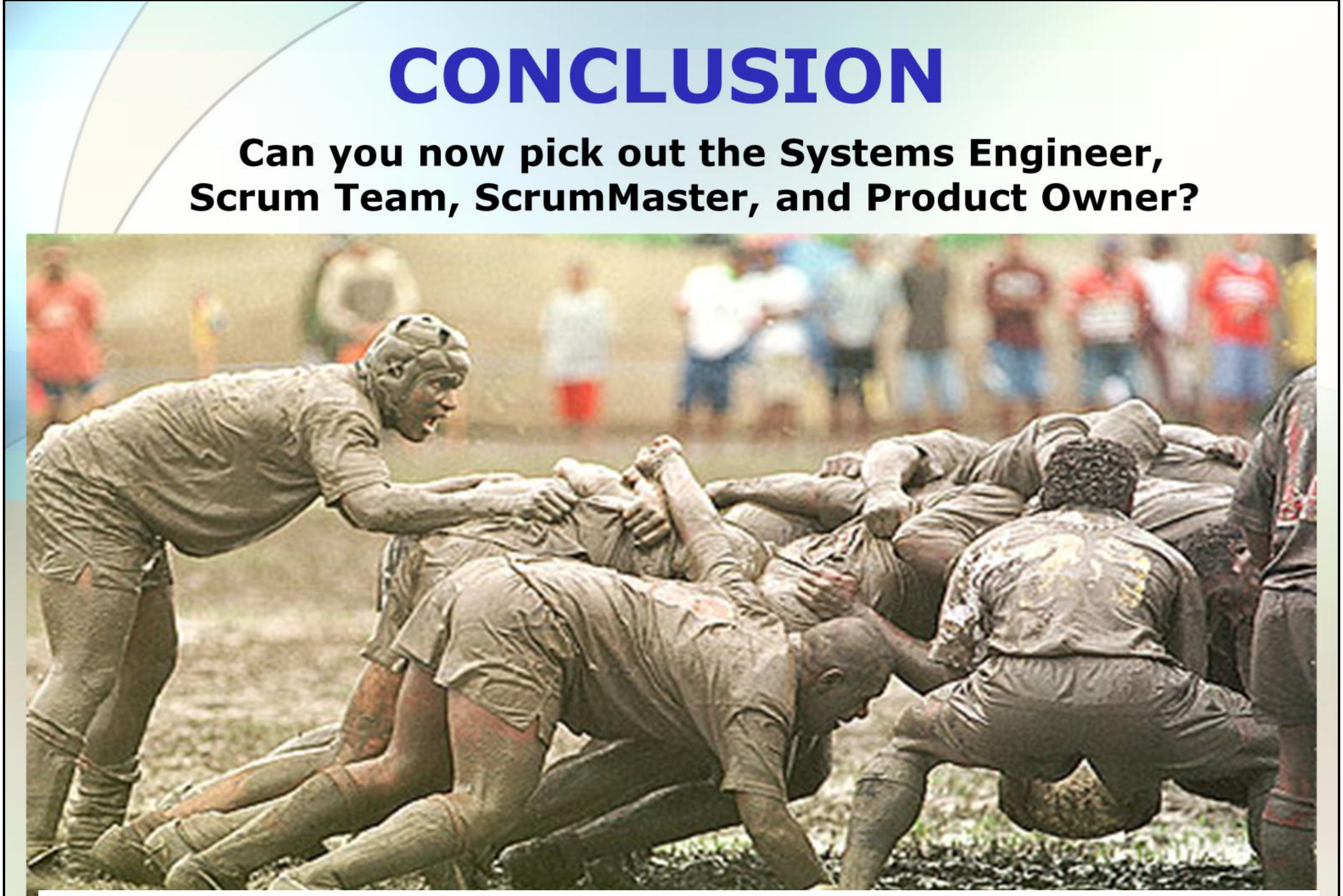
This is a picture of a typical Scrum Board used in Scrum Planning, Sprint Planning, Sprint Meetings, Iteration Meetings, etc.). It looks sort of like the one we used.

- As the Lead Systems Engineer on the Scrum Project, I was not involved in really working with this Scrum Board for the entire project.
- The SWEs and DBEs however, did all of their work off of the Scrum Board.

As far as I could tell it worked great.

- As I mentioned in my notes for the last slide. In my project the Scrum Board started out as containing almost all SE-related Stories and Tasks, to be accomplished by the SE personnel on the project team as quickly and effectively as possible.
- As the project moved along (i.e., as the SEs began turning the Needs and Requirements into Use Cases, Functions, and Features) the SWEs and DBEs began to develop from those SE task results Stories and Tasks. These SWE/DBE Stories and Tasks began to replace many of the SE stories and tasks; we had evolved into a sort of parallel Scrum Board where SEs and SWEs/DBEs had and equal number of tasks.
- The final months of the project found the Scrum Board populated with almost all SWE and DBE stories and tasks; another evolution into a Combined Scrum Board, but one that resembled a more ‘normal’ agile SW development Scrum Board.

Lessons Learned: Scrum Works. No need to avoid it. Learn it. Get in on Scrum Project Planning early. Pick up the Scrum rhythm. Have good SE Stories and Tasks ready for the Scrum Board. Keep up the Scrum pace. Stay on the Scrum Board.



CONCLUSION

Can you now pick out the Systems Engineer, Scrum Team, ScrumMaster, and Product Owner?

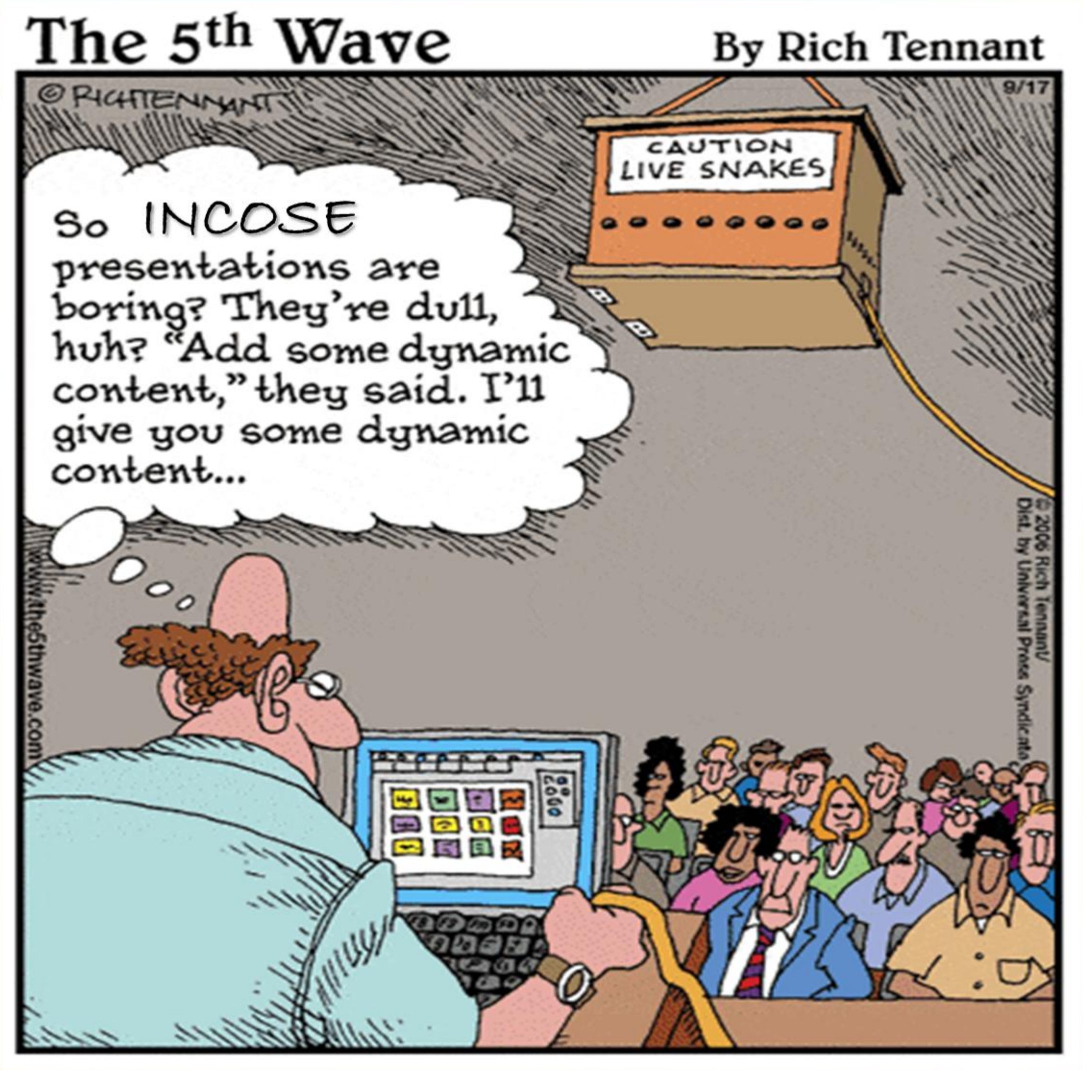
I think you understand now (one last time using the Pig and Chicken metaphor) that the gentleman on the left—contributing some directions to the Scrum , but not actually IN the Scrum—is an example of the ‘involved’ Chicken Systems Engineer.

The gentleman with the rugby ball (lower right of the graphic) is IN the Scrum And offering some concrete inputs (i.e., the ball at the right time) to the Scrum Team—but without actually influencing the Scrum Team’s direction of movement. He’s the ScrumMaster; the Facilitator; the uberPig.

You’ll see he’s also the person that the external Customer/Product Owner (the Referee barely visible on the graphic on the far right side) is paying the most attention to; the ScrumMaster is serving as direct liaison between the Scrum Team and the Product Owner/Customer Referee.

And last—but certainly not least by any means—you see in the middle of the graphic the truly committed pigs—the Scrum Team.

With contributions from the SE, and with the ball/guidance from the ScrumMaster, the Scrum Team will try to ‘move the pile’ in such a way as to accomplish what they want to vis a vis the Product Owner/Customer—A Try!


The 5th Wave
By Rich Tennant
© RICHTENNANT
9/17
CAUTION LIVE SNAKES
So INCOSE presentations are boring? They're dull, huh? "Add some dynamic content," they said. I'll give you some dynamic content...
© 2006 Rich Tennant/ Dist. by Universal Press Syndicate
www.the5thwave.com