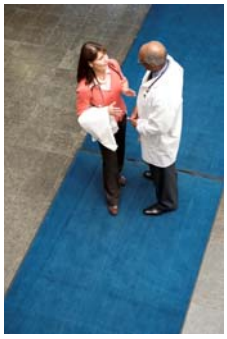
Clinical Information Systems Life Cycle and the Role of Nursing Informatics

Beverly Bell, RN, BS, MHS, FHIMSS, CPHIMS
Healthcare Executive, Premier Focus, Cerner
Beverly.Bell@Cerner.com

July 12, 2017

Questions that SDLC Can Help Answer

- What is IT building?
- How does IT know what is needed?
- Who identified the requirements?
- How do we control scope and manage change?
- Who will test it?

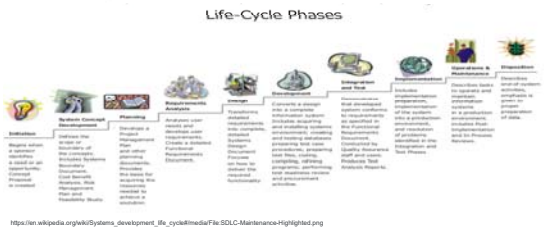


Agenda

- Objectives
- System Development Life Cycle Phases (SDLC)
- Software Development Methodologies Types
- Agile Methods
- Nursing Informatics Role
- Comments

Systems Development Life Cycle (SDLC)

- Process for planning, creating, testing, and deploying hardware and/or software
- Focuses on product requirements
  - Not a project management methodology



https://en.wikipedia.org/wiki/Systems_development_life_cycle#/media/File:SDLC-Maintenance-Highlighted.png

Objectives of Presentation

- State the purpose of System Development Life Cycle (SDLC)
- Compare differences between Agile and Waterfall
- Name 2 variations of Agile methodologies

Theories may be used by your hospital IT department

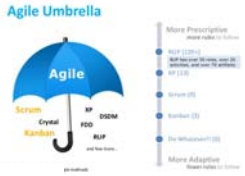
Software Development Methodologies

- Life cycle approaches to systems development evolve
  - Technology constant changing
    - Compile code hours => seconds thus validation immediate
  - How we communicate and place of work
    - mobile, multiple locations, restaurants
  - Tools use
    - Texting, MicroSoft Snip It
  - Work changing
    - Software written by engineers that used the software
    - Many experts involved today, collaborate together
  - Overcome the inherent deficiencies
- Types of SDLC
  1. Incremental / adaptive
  2. Iterative
  3. Sequential / predictive
  4. Hybrid / anamorphic

www.torak.com

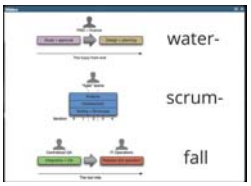
Software Development Methodologies

- 1. Incremental / Adaptive
  - Scrum
  - Kanban
- Focus on team collaboration, self-organizing teams
- Each has different points of emphasis
  - Select what works / intermix
  - Scrum story / Kanban board



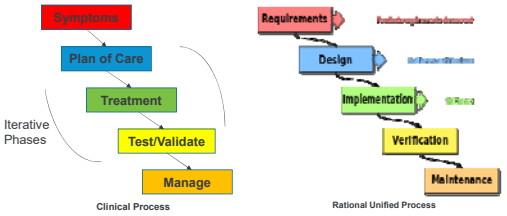
Software Development Methodologies

- 4. Hybrid / Anamorphic
  - WaterSCRUMfall
    - Waterfall to clarify the front end
      - Vison, plan, budget
    - Agile for the developers
    - Return to waterfall once product near release
  - EMR implementations have been done in this manner



Software Development Methodologies

- 2. Iterative methodologies
  - Rational Unified Process (RUP)
- Process framework, phases, tool driven, artifact heavy
- Keep fine tuning the product over multiple reiterations



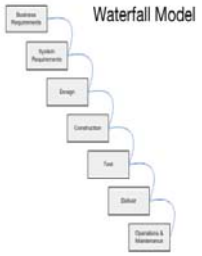
Agile Foundational Values

- 2000 created, 17 thought leaders/pioneers
- Agile Manifesto is comprised
  - Twelve (12) supporting principles
  - Four (4) foundational values
    1. Individuals and interactions over processes and tools
    2. Working software over comprehensive documentation
    3. Customer collaboration over contract negotiation
    4. Responding to change over following a plan
- Each Agile methodology applies the values and principles in different ways

https://www.smartsheet.com/comprehensive-guide-values-principles-agile-manifesto

Software Development Methodologies

- 3. Sequential / Predictive
  - Waterfall (Classic)
- 1956 manufacture, construction
- Big design upfront
- Phases are sequential
- Get it right the first time or start from the beginning
- Challenges
  - Designers unaware of future difficulties
  - Clients change requirements
- EMR designed upfront 12-18 months before go live, features not relevant at go live
- Nurse optimization projects



Agile Principles

1.	Customer satisfaction through early and continuous software delivery
2.	Welcome changing requirements throughout the development process
3.	Frequent delivery of working software with preference to shorter timescale
4.	Collaboration between the business stakeholders and developers throughout the project. Must work together daily
5.	Build projects around motivated individuals. Give them the environment and support and trust them to get the job done
6.	Enable face-to-face conversation
7.	Working software is the primary measure of progress
8.	Agile processes to support a sustainable and consistent development pace
9.	Attention to technical excellence and good design enhances agility
10.	Simplicity – maximize the work not done is essential
11.	Self-organizing teams encourage great architectures, requirements, and designs
12.	Regular reflections on how to become more effective, then tune and adjust behavior accordingly

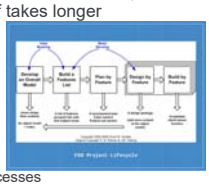
https://www.smartsheet.com/comprehensive-guide-values-principles-agile-manifesto

Challenges Using Agile

- Defining the problem
- Aligning vision with interaction
- Clarifying the role of the Product Owner
- Not building in real customer feedback loops
- Developing a "WaterScrumFall" process
- Losing the forest for the tree

Agile: Feature-Driven Development (FDD)

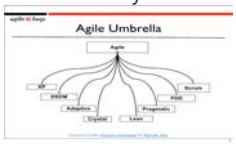
- 1997 built around software engineering best practices
  - E.g. domain object modeling
  - Focus on design and build features
  - Unlike other Agile methods, it describes very specific and short phases of work that has to be accomplished separately per feature
  - A feature cannot take longer than two weeks to build, will be broken down into smaller features if takes longer
- Five (5) phases
  1. Development of an overall model
  2. Building of a feature list
  3. Planning by feature
  4. Designing by feature
  5. Building by feature
     - Last 2 phases are short iterative processes



https://project-management.com/xp-fdd-dsdm-and-crystal-methods-of-agile-development/

Agile Family of Methodologies

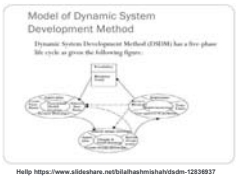
1. Short, timeboxed periods called sprints
2. Self-organizing teams develop working features that are added in small iterations to the software
3. Emphasis on communication, collaboration and flexibility to adapt to emerging business realities
4. Continuous integration of development and testing throughout the software development
5. Delivers working tested software every iteration



https://project-management.com/xp-fdd-dsdm-and-crystal-methods-of-agile-development/

Agile: Dynamic Systems Development Method (DSDM)

- 1995 published by the DSDM Consortium
  - Best practice experiences of people working in big companies; British Airways, American Express, Oracle
  - Five (5) phases
  - Focus on business need
  - Small teams are empowered to make decisions
  - Useful if need to develop systems in short time



https://project-management.com/xp-fdd-dsdm-and-crystal-methods-of-agile-development/ Help https://www.slideshare.net/bilalhashmishahi/dsdm-12838937

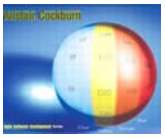
Agile: Extreme Programming (XP)

- 1996 created to take into account changes that were happening
  - E.g. object oriented programming and the Internet
- Take best practices to "extreme levels"
  - E.g. instead of nailing down requirements => constant customer feedback to eliminate defects early
- Places importance of co-location of customers and developers as well as pair programming (share code knowledge)
- Focuses on frequent, 14 day development cycles
- Helpful when constant changing requirements or when not sure about functionality of the system
- Six (6) phases
  1. Planning
  2. Analysis
  3. Design
  4. Execution
  5. Wrapping
  6. Closure

https://project-management.com/xp-fdd-dsdm-and-crystal-methods-of-agile-development/ http://www.guru99.com/agile-scrum-extreme-testing.html

Agile: Crystal

- Developed by Alistair Cockburn
  - Studied and interviewed teams
- Methods are color-coded according to the number of people being coordinated
  - Clear for 8 or fewer people
  - Red is for 50-100 people
- Seven (7) common properties in Crystal
  1. Frequent delivery
  2. Reflective improvement
  3. Close communication
  4. Personal safety (first step in trust)
  5. Focus on solution development
  6. Easy access to expert users and technical environment
- Methods are very flexible and avoid rigid processes due to the human focus



Crystal Clear: A Human Powered Methodology for Small Teams by Alistair Cockburn 1998

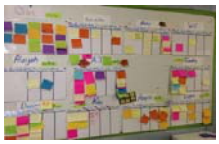
Agile: Lean Software Development

- Based on the principle just in time production
- Translation of Lean manufacturing and Lean IT principles to software development
- Aims at increasing speed of software development and decreasing cost
- Seven (7) principles
  1. Eliminating waste
  2. Amplifying learning
  3. Defer commitment (deciding as late as possible)
  4. Early delivery (deliver as fast as possible)
  5. Empowering the team
  6. Build integrity in
  7. See the whole

http://www.guru99.com/agile-scrum-extreme-testing.html, https://en.wikipedia.org/wiki/Lean_software_development

Agile: Kanban

- Four (4) core principles
  - Visualize work
    - Create a card to see your work effort
    - Also see bottlenecks, challenges
  - Limit work in process
    - Avoid problems caused by task switch
    - Reduce need to constantly reprioritize
  - Focus on flow
    - Use work in process limits and team driven policies to improve smooth flow of work
  - Continuous movement
    - Measure effectives by tracking flow, quality, throughput, and lead times



https://project-management.com/xp-fdd-dsdm-and-crystal-methods-of-agile-development/

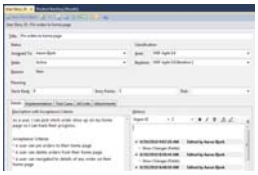
Agile: Four (4) Aspects of Lean Manufacture

- **Flow**
  - Base your management decisions on long-term philosophy even at the expense of short term financial goals
  - Continuous Process Flow (CPF) to bring problems to the surface (previous step flows into next step with zero time between steps)
  - Use pull systems to avoid overproduction (waste)
  - Level out workload (slow and steady)
  - Build a culture of stopping to fix problems, to get quality right the first time
- **Visualization of work**
  - Create representation of workflow
  - Use visual controls so no problems are hidden – entire team see
- **Grow Leaders and Team**
  - Grow leaders who thoroughly understand the work, live the philosophy and teach it to others
  - Develop exceptional people and teams who follow your company's philosophy
  - Respect your extended network of partners and suppliers by challenging them and helping them improve
  - Once decision made, implement rapidly. Support environment for knowledge workers to be successful and let them figure it out
  - Implement cross functional teams (e.g. architects, engineers)
- **Learning organization**
  - Become a learning organization through relentless reflection and continuous improvement

www.leanblitzconsulting.com

Agile: Scrum Methodology

- Single user story goes through SDLC phases in a single 2 week sprint
  - Multiple user stories coincide with multiple sprints during the development project
  - Product Owner writes a user story for a customer requested feature then stories are rolled up into multiple sprints



Sprint #1

- User Story #1 — As a user_I should_so I can…
- User Story #2 — As a user_I should_so I can…
- User Story #3 — As a user_I should_so I can…

Sprint #2

- User Story #1 — As a user_I should_so I can…
- User Story #2 — As a user_I should_so I can…

http://www.yodiz.com/blog/what-is-epic-in-agile-methodology-definition-and-template-of-epic/

Agile: Kanban

- 1940s, Toyota picked up idea from grocery stores as attendants used cards to restock shelves
- Japanese word, a card containing all information needed to be done on the product at each stage along its path to completion
- Uses a visual card or signboard that displays
  - different stages of development
  - number of work items to do, being worked on, and completed by person
- Uses incremental improvements to continuously add features until the product is fit for delivery
- 2005, used to implement Lean and Agile methodologies in technology

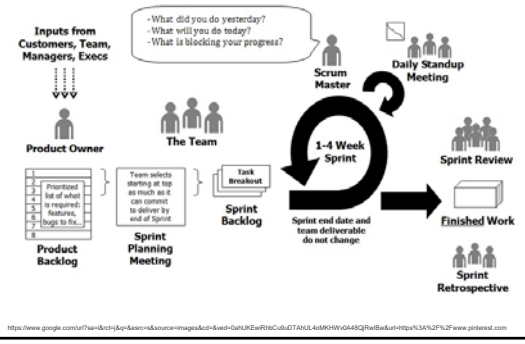
https://project-management.com/xp-fdd-dsdm-and-crystal-methods-of-agile-development/

Agile: Scrum Methodology

- Three (3) roles
  - Product Owner
    - Person with vision, authority, and availability
    - Responsible for continuously communicating vision to the development team and answer questions from the team
    - Meets with customer (frequent validation with customer)
  - Scrum Master
    - Acts as a facilitator for the Product Owner and the team
    - Conducts daily stand up meeting at start of workday
    - Does not manage the team (not personnel manager)
    - Removes any impediments that are obstructing the team from achieving its sprint goals
    - Advise the Product Owner about how to maximize ROI for the team
  - Cross Functional Team
    - Software engineers, architects, programmers, analysts, quality assurance experts, testers, and user interface designers
    - Consists of 3-9 persons, ideally in one room, free of distractions
    - Responsible for self organizing to complete work

http://scrummethodology.com/

Agile: Scrum Methodology



Comments / Thoughts / Questions

Nurse Informatics Role

- Customer or customer proxy
  - Understand requirements, workflows and processes
- Domain expert
  - Understand the system design and/or build
  - Knowledgeable of how system works
- Cross functional team participation
  - Collaborate with engineers, programmers
  - Balance customer need with technical constraints
- Optimization of EMRs
  - 6-9 months vs 15-18 month projects
  - Identification of areas to optimize based on data

Nurse Informatics Role

- New apps/software being developed
  - Lead
  - Innovate
  - Design
  - Participate
    - Ask questions, take an interest
- Communicate to nursing community
- Certify in Agile