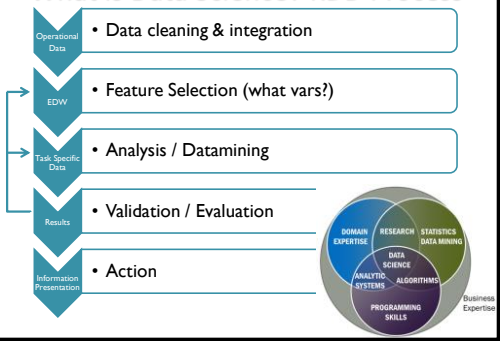
Take Away I
What is Data Science? KDD Process



Take Away II
What is Big Data ?

- 4 Vs of Big Data
  - Volume : lots of data
  - Velocity : constantly generating & changing
  - Variety : expressed in many ways
  - Veracity : lots of errors
  - (Value)
- Big Data Problems
  - Time
  - Money
  - Quality (Precision)
- Three Resources: AMP
  - Algorithm
  - Machine
  - People

Take Away III
Business uses of data mining

- Essentially five tasks…
- Classification
  - Classify credit applicants as low, medium, high risk
  - Classify insurance claims as normal, suspicious
- Estimation
  - Estimate the probability of a direct mailing response
  - Estimate the potential cohort size for a clinical trial
- Prediction
  - Predict which customers will leave within six months
  - Predict which patient will return to the ED
- Affinity Grouping
  - Find out what books to recommend to Amazon.com users
  - Find treatment regime that was successful for similar patient
- Description
  - Help understand large volumes of data by uncovering interesting, useful, and actionable patterns

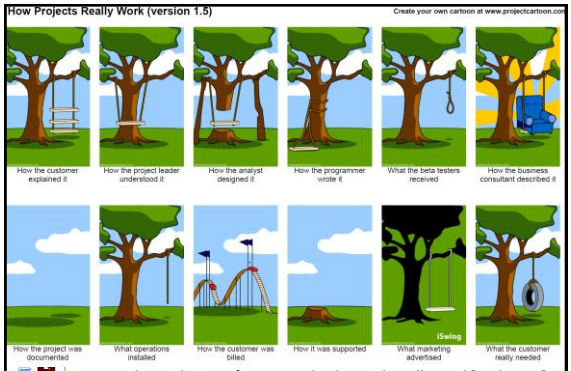
Take Away: Coding
Standard billing forms & codes

- Know "how to do" coding
  - examples from class
- Codes (look at handouts)
  - ICD-10-CM
  - ICD-10-PCS
  - CPT

Managing Software Needs (Acquisition and Development)

Agenda

- Why effective communication is so important
- Approaches to meeting SW needs
- Facts & Fallacies of SW development
- Case example
  - Agile: Scrum



- Whose job is it to figure out what hospitals really need ? Who can?
- http://projectcartoon.com/cartoon/2

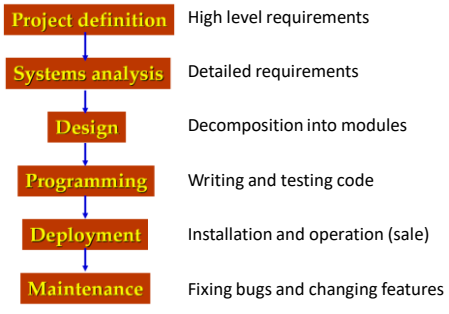
Agenda

- Why effective communication is so important
- Approaches to meeting SW needs
- Facts & Fallacies of SW development
- Case example
  - Agile: Scrum

Meeting Software Needs
Approaches to software development

- Traditional systems development life cycle
- Prototyping
- Packaged software: off the self
- End-user development: in-house
- Outsourcing
- Open source
- SaaS: Software as a Service

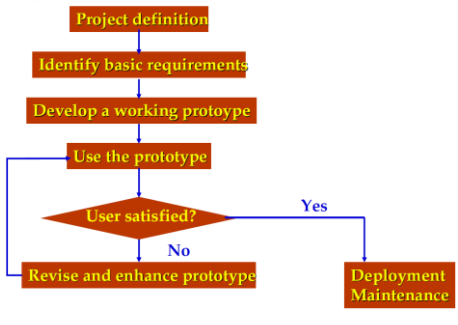
Traditional systems development life cycle ("waterfall" model)



Traditional systems development life cycle ("waterfall" model)

- Advantages
  - For well-understood problems, produces predictable outcomes
- Disadvantages
  - Inflexible
  - Long delay before any useful results
    - May be obsolete by then
  - Often hard to know requirements until actual use

Prototyping ("Iterative" model)
Agile Method



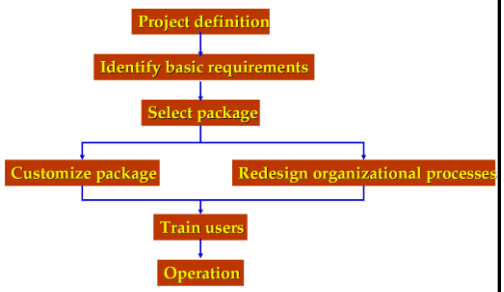
Prototyping ("Iterative" model) Agile Method

- Advantages
  - Especially useful when exact requirements are hard to know in advance
    - user interfaces
    - decision systems
    - electronic commerce?
  - Encourages user involvement
- Disadvantages
  - Hard to predict and control outcomes reliably
  - If repeated, significant reimplementations are needed, can be very expensive
  - May result in systems that are inefficient, unreliable, or hard to maintain

Agile Manifesto

- Philosophy
  - Adaptive vs predicative
  - Iterative vs waterfall
  - Code vs documentation
- Scrum: Do sprints
  - sprints are short two-to-three week design or development cycles via a repeatable process where the team works on designing and/or developing specific user stories. Depending on the size of the application being built, there may be many sprints. But the rapid iterations move the project forward quickly and allow the team to focus on the needs of the end user.
- Extreme programming, lean software development, test driven
- prototyping

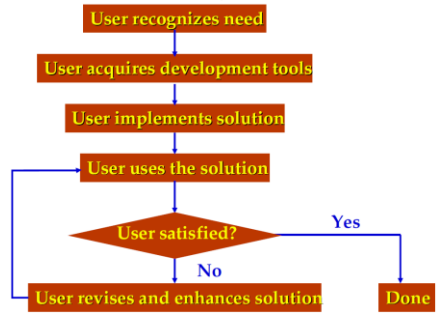
Packaged software



Packaged software

- Advantages
  - By amortizing development and maintenance costs over many organizations, it is possible to get superior solutions at much lower cost
- Disadvantages
  - Customizing software can be very time-consuming and expensive
  - May have to change organization to fit software, rather than vice versa

End-user development (in house development)



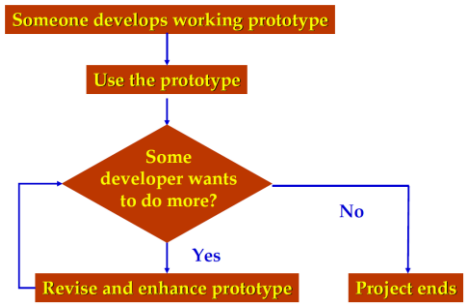
End-user development (in house development)

- Advantages
  - Can be *much* faster
  - Improved requirements determination
  - Increased user involvement and satisfaction
- Disadvantages
  - Often, users lack the right implementation skills
  - Many problems can't be solved within the limitations of the tools
  - Lack of quality assurance and standards for programs and data
  - Lack of sharing of programs and data
  - Reduced opportunity for reuse of results

Outsourcing

- Contract out the performance of any or all of the above steps to another firm
- Advantages
  - Economies of scale
  - Flexibility
  - Predictability
  - Freeing up human resources and capital
- Disadvantages
  - Loss of control
  - Vulnerability of strategic information
  - Dependency

Open source



Open source

- Advantages
  - Usually lower cost
  - Sometimes easier to adapt "packaged" software to own needs
  - "Philosophically" appealing to many people
- Disadvantages
  - Usually lower quality support
  - Only a few kinds of software are currently available in this format (Linux operating system, Apache web server, etc.)

Software as a Service (SaaS)

- Cloud based: gmail
- Advantages
  - Do not have to maintain hardware/software
  - Economies of scale
  - Predictability
  - Freeing up human resources and capital
- Disadvantages
  - Loss of control
  - Dependency
  - No access when network is down
  - In the long run expensive

Agenda

- Why effective communication is so important
- Approaches to meeting SW needs
- Facts & Fallacies of SW development
- Case example
  - Agile: Scrum

Problems with software development

- Computerworld magazine
  - "Nearly one-third of all projects fail"
  - "More than half come in over budget"
  - "Only 16% of all projects come in on time and on budget"
  - Survey of 8000 projects from 385 companies.
- Key factor for success or failure:
  - "User involvement/input"

Facts about Software Development

- Facts
  - The most important factor in software development is the quality of the programmers.
  - The best programmers are up to 28 times better than the worst.
  - Adding people to a late project makes it later.
  - One of the most common causes of runaway projects is poor estimation.
  - The other most common cause of runaway projects is unstable requirements.
  - Requirements errors are the most expensive to fix during production.
  - Maintenance typically consumes 40 to 80 percent of software costs.
  - Enhancements represent roughly 60 percent of maintenance costs.

Adapted from Robert L. Glass, *Facts and Fallacies of Software Engineering*, Addison Wesley, 2003

Fallacies about Software Development

- Fallacies
  - Software needs more methodologies.
  - You teach people how to program by showing them how to write programs.

Adapted from Robert L. *Glass, Facts and Fallacies of Software Engineering*, Addison Wesley, 2003

Agenda

- Why effective communication is so important
- Approaches to meeting SW needs
- Facts & Fallacies of SW development
- Case example
  - Agile: Scrum

The following slides are adapted from http://www.trizetto.com/HCC2014/

TriZetto 2014 HEALTHCARE CONFERENCE
Competing to Win in a Post-Reform World

CareAdvance Provider: Agile Development Collaboration

Janice Mead, RN Sr. Product Manager
CareAdvance Provider Product Owner

MAY 18 – 21, 2014
JW Marriott San Antonio Hill Country

TRIZETTO

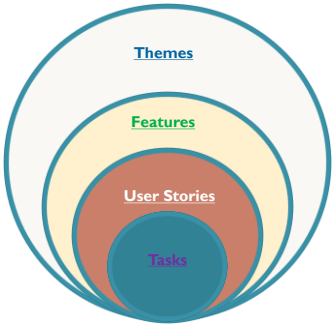
CareAdvance Enterprise Agile Roles

Product Manager
Product Owner
Scrum Master
Scrum Team

Product Backlog Hierarchy

- Theme
  - An idea that spans multiple Scrum Teams and Sprints
  - May fit into a Release
  - No detail
- Feature
  - An idea that may span multiple Scrum Teams and Sprints
  - Usually fits into a Release
  - Some detail
- User Story (a.k.a. Product Backlog Item)
  - Work item for a Sprint for a Scrum Team
  - Complete testable piece of work
  - Very detailed
- Task
  - Chunk of work for team members
  - When all tasks are completed and reviewed, the User Story is done

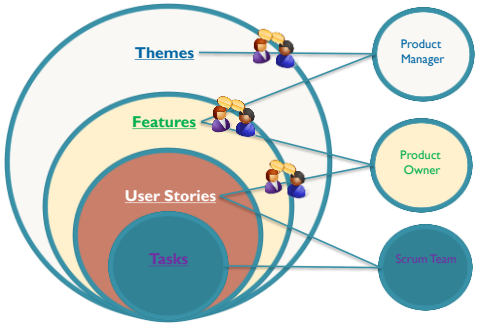
Product Roadmap Hierarchy



CAE Agile Roles

- Product Manager
  - Defines Roadmap - High Level Themes and Features
  - Presents business need and value to the Product Owners
- Product Owner
  - Writes User Stories for Features
  - Prioritizes the User Stories in the Backlog
  - Grooms the User Stories with the Scrum Team
- Scrum Team Members
  - Grooms and Sizes the User Stories
  - Commits to doing the work
- Scrum Master
  - Protects the team
  - Removes impediments to completing work

Product Roadmap Ownership



CAE Scrum Teams

- Clinical CareAdvance (CCA)
  - Multiple Scrum Teams– logical areas of responsibility
    - Case/Disease Management
    - Utilization Management
    - Application Workflow
    - Constituents
    - Content
    - Rules
    - Common Application Components
    - Infrastructure
    - Development Operations and Architecture

CAE Scrum Teams

- CareAdvance Provider
  - One Team responsible for the Product
- Value Based Benefits
  - One Team responsible for the Product
- Each Scrum Team has its own backlog
  - Product Owners identify inter-team dependencies
  - Product Owners set the priority for their team's backlog
  - Team and PO groom User Stories to remove ambiguity and size the effort
  - Team makes the Release and Sprint commitments

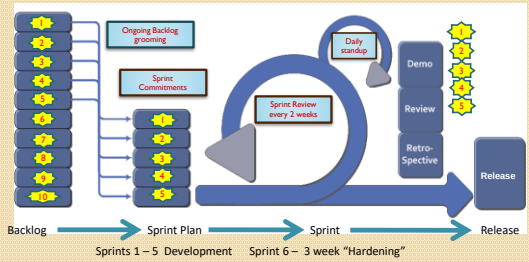
Client Collaboration

[Health care provider collaborating with engineers]

When

Who

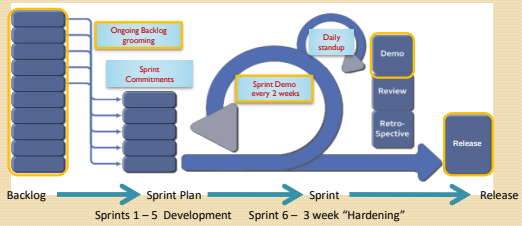
Agile Development



- Adding Features and grooming the backlog is an ongoing activity.
- User Stories are worked on in backlog priority order during Sprints.
- "Done" work is shown to stakeholders at the end of each Sprint.

Opportunity for Client (=health care providers) Influence

- Backlog Grooming
  - **Product Manager Defines Themes and Features**
  - **Product Owner adds User Stories to the Backlog; Prioritizes**
- Team Grooming and Commitments
- Daily Stand-up
- Sprint Demo
- Release Review



Client Opportunities for Input

- Theme and Features – Think Breadth
  - What is the business need?
  - What problem are you trying to solve?
  - What is your high level workflow?
- User Stories – Think Depth
  - What does the user need to do and why?
  - What is acceptable functionality?
- Sprint Demo
  - How does what we built so far look?
  - Are we headed in the right direction?

End-of-Sprint Demos

- Teams show off completed work
  - Not a technical review
  - Only pieces of the full picture
- Stakeholders are encouraged to attend
- Opportunity to give feedback
  - Fail fast to allow time to respond
- Work shown may or may not be in the release
  - Complex feature work may stretch into the next release

Take Away 1
Approaches to software development

- Traditional systems development life cycle
- Prototyping
- Packaged software
- End-user development
- Outsourcing
- Open source
- SaaS: Software as a Service

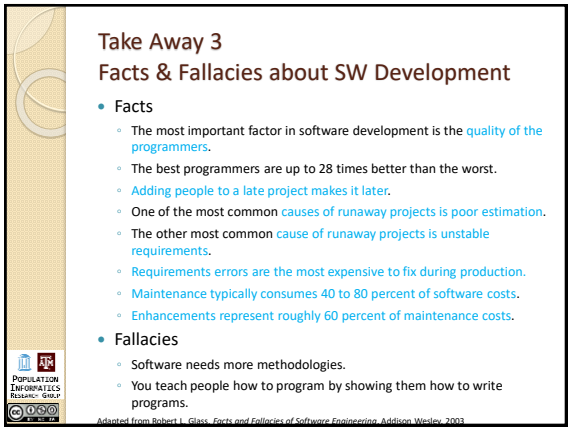
Take Away 2
Traditional (waterfall) vs Agile (iterative)

Waterfall (Traditional)

- Know the process
- Advantages
  - For well-understood problems, produces predictable outcomes
- Disadvantages
  - Inflexible
  - Long delay before any useful results
    - May be obsolete by then
  - Often hard to know requirements until actual use

Iterative (Agile)

- Know the process
- Advantages
  - Especially useful when exact requirements are hard to know in advance
    - user interfaces
    - decision systems
    - electronic commerce?
  - Encourages user involvement
- Disadvantages
  - Hard to predict and control outcomes reliably
  - If repeated, significant reimplementations are needed, can be very expensive
  - May result in systems that are inefficient, unreliable, or hard to maintain


Take Away 3
Facts & Fallacies about SW Development
Facts
The most important factor in software development is the quality of the programmers.
The best programmers are up to 28 times better than the worst.
Adding people to a late project makes it later.
One of the most common causes of runaway projects is poor estimation.
The other most common cause of runaway projects is unstable requirements.
Requirements errors are the most expensive to fix during production.
Maintenance typically consumes 40 to 80 percent of software costs.
Enhancements represent roughly 60 percent of maintenance costs.
Fallacies
Software needs more methodologies.
You teach people how to program by showing them how to write programs.
Adapted from Robert L. Glass, Facts and Fallacies of Software Engineering, Addison Wesley, 2003
Population Informatics Research Group