Will Kelsey

APC Prof. Carpenter

Summer 2019

CURSE Process Models

Waterfall Model

1. Specifications
   1. Database of users
      1. ~100 students
      2. 10 instructors
      3. 1 admin
   2. Database of courses
      1. CRN
      2. Course name
      3. Times
      4. Instructor
   3. Students
      1. Can register
      2. See available courses
      3. See own schedule
   4. Instructor
      1. See available courses
      2. See own course roster
   5. Admin
      1. Can see all
      2. Can edit all
         1. Users
         2. Courses
         3. Schedule
         4. Etc.
   6. Includes multiple semesters
   7. Printable schedule
   8. Scheduling preferences
2. System Architecture
   1. Hosted server
      1. Ability to handle all requests
      2. Load balancing for peak usage times
         1. Registration day
      3. Admin portal for administering local
      4. Redundant backups
   2. Web clinet
      1. User viewable
      2. Client side display
      3. Web data view only
3. Implementation and Unit testing
   1. Software development
      1. Software creation for server side
   2. Server creation
      1. Hardware config and redundancy
   3. Creative team on web interface
      1. Visual bugs and accurate access
4. Integration and system testing
   1. Bring in sample data
   2. Use beta testers for single class
      1. Maintain previous system as backup
      2. Close communication
5. Operation/Maintenance
   1. Onsite personnel
      1. Rapid repair of bugs and issues for short time after rollout
   2. Global support team
      1. Support multiple instances on many sites

Incremental Development Model

1. Specifications
   1. Database of users
      1. ~100 students
      2. 10 instructors
      3. 1 admin
   2. Database of courses
      1. CRN
      2. Course name
      3. Times
      4. Instructor
   3. Students
      1. Can register
      2. See available courses
      3. See own schedule
   4. Instructor
      1. See available courses
      2. See own course roster
   5. Admin
      1. Can see all
      2. Can edit all
         1. Users
         2. Courses
         3. Schedule
         4. Etc.
   6. Includes multiple semesters
   7. Printable schedule
   8. Scheduling preferences
2. First Revision
   1. Basic functionality
      1. Pull requests from clients
      2. Research based
   2. Develop interest from real world
   3. Could be course schedule first
3. Second Revision
   1. Build out with specs from research
   2. Use client feedback
   3. Scheduling with course instructor database
   4. Begin development of web interface for local program
4. Third Revision
   1. Server based
      1. Rudimentary access to a server
      2. Limited control
   2. Student database with student schedule
5. Fourth Revision
   1. Web interface
      1. Focus on student access
   2. Student access to viewing courses and professors
   3. First instance of registration system for student side
6. Fifth Revision
   1. Web interface makeover
      1. Focus on design and ease of use
   2. Full implementation of student course registration
   3. Begin use of student preferences
      1. Time
      2. Professor
      3. Elective preference
7. Sixth Revision
   1. Full automation of course scheduling based on student preferences
   2. Automatic efficient scheduling for course timing and locations
8. Further development based on customer requests

Integration Model

1. Specifications
   1. Database of users
      1. ~100 students
      2. 10 instructors
      3. 1 admin
   2. Database of courses
      1. CRN
      2. Course name
      3. Times
      4. Instructor
   3. Students
      1. Can register
      2. See available courses
      3. See own schedule
   4. Instructor
      1. See available courses
      2. See own course roster
   5. Admin
      1. Can see all
      2. Can edit all
         1. Users
         2. Courses
         3. Schedule
         4. Etc.
   6. Includes multiple semesters
   7. Printable schedule
   8. Scheduling preferences
2. Pick a platform to jump from
   1. CourseStorm
   2. Regpack
   3. ProClass
   4. RegFox
   5. Aventri
   6. Amilia
   7. Learning Sdtream
   8. ePly
   9. RedPodium
   10. EventBank
   11. Eventbrite
   12. Eventgrid
   13. Race Entry
   14. Pike13
   15. Arlo
   16. Configio
   17. 123Signup Association Manger
   18. InitLive
   19. Bizzabo
   20. RegOnline
   21. <https://www.softwareadvice.com/registration/>
   22. <https://www.capterra.com/class-registration-software/>
3. Choose
   1. Price
      1. Costs per head
      2. Per class
      3. Per institution (lump)
      4. Subscription
      5. Limit count pricing
   2. Platform
      1. Linux
      2. Windows
      3. Mac (prefer alternatives; see Linux)
   3. Size of capacity
      1. Small
      2. Medium
      3. Large
      4. Enterprise
      5. Consortium
   4. Ease of development
      1. Community support
      2. Trainings
      3. Onsite support
      4. Preconfigured by default
4. Implement
   1. Depending on service/services chosen
   2. Configure for local environment
   3. Build up application specific features
5. Testing and patching
   1. Start with small sample and beta testers
   2. Use support if needed
6. Deployment