Agile Design with Micro Services
(RentCom)

Background

Who am I? Find me on LinkedIn: https://www.linkedin.com/in/allenedixon/

What is RentCom? RentCom is an application I developed for my fathers business. The problem was that software for commercial rental business was very expensive back in the 80's. The business was in a very small town. My father had looked a several COTS applications and found the difficult to use as well. It amazes me today how many point of sale (POS) applications are so out of date with current technology and usability even today.

The business started renting tools and party goods. In 1984, my father decided to add movie and VCR rentals to the fix. This was not a problem for hand written invoices because the volume of rentals was low. Well, he had 100 movies and 5 VCRs. RentCom version 1.0 was a movie database on a Commodor 64. But, by 1990, the business was struggling with the volume of hand written invoices, decrease in the video rental rates, 2000 movie inventory and increased competition. Buy this time, movies where stored by category, sorted with a bubble sort and printed. A new movie list took 2 days to print. It seem like every gas stations and supermarket was renting videos.

After developing software for 5 years using structured analysis and design methodologies, I decided to remove the old cash register and hand written invoices. The business needed to move through more customers with less labor.

RentCom version 2.0 was born. It was a single tier application written in FoxPro 5. I work with my father to develop a POS application on MS Windows 3. We iterated over the design and development over 2 months introducing functional highest value first. I didn't know at the time that the fast software development methodology would become popular in the future. That was the birth of Agile development for me. Wondered why we didn't us it in the industry. FoxPro provided firm utilities to build UI and design a database.

RentCon version 3.0 was introduce a few years later after a power surge sent the x286 to retirement. With Windows 95, FoxPro 6.0 and 386 computer, the application became easier to us with more functionality.

Fast forward to 2003, n-tier applications are all the rage. My father's rental business no longer rents VHS tapes or tools. There is an inventory of over 500 DVD movies, party rental and retail. The rental prices continue to plunge on movies. Enter RentCom 4.0, client server, MySQL database server, Java Swing, bar code readers, and 3 computers for cash registers.

In the year 2012, there is no more DVD and BluRay movies in the inventory. Movie streaming is all the rage. It costs more than renting a BluRay movie. But, there is not late fee. The movie is always in stock. The doors close on the brick and mortar business. Enter the age of internet companies. No my father still has his business. It could be the most successfully internet business in the county. Maybe the only internet business. The party rentals are still going strong. There is no rent that needs to be paid. Well not rent for a building. RentCom 5.0 is located at http://gunnisonparty.com/. Enter Saas application. Okay, this web site is a simple static HTML 5 application.

The reason I went through the history of the family business is to show that technology is always changing and that I am qualified to be the Domain Expert, Software Architect and Developer for

RentCom 6.0.

Purpose

RentCom 6.0 may never be release as a product. The purchase is to demonstrate how to build an SAAS application using DevOps, Agile, TDD, and PAAS. Why, because technology and business does stand still. Software Architects need to be continuously learning even if their company does not provide the time during working hours.

Thanks goes out to Eric Evans the father of Domain Drive Design, Vaughn Vernon for helping me understand DDD better, and Moise Macero the practical developer for his help with Micro Services. For information on DDD, TDD and Micro Services, purchase their books.

Concept

All projects start of as concept. As an architect, it is important to design an initial functional, technical and deployment architecture. This may appear to be waterfall process. The key word here is “initial”. I will iterate over the architect in an Agile method similar to code development.

RentCom 6.0 will go back to its roots; tool rental and party goods. Movie rental would require its own bonded context and services. I don't believe BluRay rentals can compete with RedBox and movie streaming. Since RentCom 6.0 is targeted to be Software as a Service, it needs to support more than one company. Some rental companies have more than one store. Looking forward, I believe tool and party rental could be successful without a store front. There is proof Gunnison Party does not have a brick and mortar store. The inventory is stored in sheds and a garage. The product is delivered to the customer, set up and removed by Gunnison Party. This is likely the future for tool rentals. So, RentCom will support stores and online tool rental. This is the beginning of the functional requirements.

Systemic requirements are important to define in the concept phase of the project. Expect the systemic requirements to change. The systemic requirements impact that technical and deployment architectures.

Systemic Requirements:

Category	Requirement
Availability	The application with have 99.999% up time accept for scheduled downtime for major upgrades. The clause in red protects you in the event downtime is required. Since RentCom will be used in North America only, rental companies are normally open 10 to 6, late nights can be used for a major upgrade.
Availability	The application will remain available when services and client changes are made.
Usability	The location will support localization because it is used in Canada. (future)
Usability	Future requirement: Many people us Spanish as their primary language. Supporting Spanish for the online client.
Scale ability	The application needs to scale to support ever growing number of users and companies.

Category	Requirement
Security	The application needs to authenticate users.
Security	Finance transactions and personal data will be store. The data needs to be secured.
Liability	Need to have a legal privacy statement for when users sign up for rentals.

Notice that systemic requirements end in ility. They are also know as ilities.

Initial Functional Architecture

There are many tools to document a functional architecture. 3DM uses a 5 tier functional mapping model. OOD methodologies like the unified process has use cases, UML, state diagrams,... These functional architectures require more of a water fall approach. I am a certified scrum master. So, I will be using Scrum and XP tools.

Since I will be building Micro Services, I will be using some DDD tools as well. I thought it best to start with Event storming. The file Event Storm Day 1.pdf contains the artifact produced from the 1st day. I project will likely have more than one day of event storming.

The second functional architectural artifact is the Context Diagram from Day 1.pdf.

You will note that the number of Bonded Contexts has been reduced to 3. This is a result of refactoring. The Store Context appears to be large. Since this is an Agile project, the Bonded Contexts and Context maps will change has more information is presented by the Domain expert.

Initial Technical Architects

Following the Micro Services Characteristics:

- Small in size
- Messaging enable
- Bonded by contexts
- Autonomous developed
- Independently deploy-able
- Decentralized
- Built and released with automated processes

I have constructed my initial technical architecture. Concept Technical Architecture.pdf

Initial Deployment Architecture

Form the systemic requirements already listed in this concept. It is clear we need a platform as a service (PAAS). For this application I have chosen Amazon Web Services (AWS) and docker contains. This will allow the application to scale up and down as need. You will note that there is not a deployment diagram. The PAAS solution (AWS) will provide the view on the deployment architecture. The solution will scale up and down on virtual servers. A diagram will be quickly out of date. The technical diagram will provide the necessary view into the solution.