# **1 Business Overview and Processes**

The Drexel County General Hospital (DCGH) located in Philadelphia has been updating their systems to support the increasing number of patients after recent population growth in the surrounding area. In the past, the hospital relied on physical record keeping, but this method has become too cumbersome and time consuming for the staff to quickly process information for their patients. The hospital has been migrating physical information such as appointments, patient info, and prescriptions to a digital system. The development efforts include creating a back-end system to store and input old records as well as creating new front-end software (website/app) for the staff and patients.

The existing work is composed of two separate teams, the IT team responsible for the back-end and Software Development team responsible for the front-end. Each team works separately on their capabilities with integration events to work through interface issues. They are each led by their own project manager with independent schedules. Of the two teams, the IT group has a longer legacy at the hospital as they started overhauling the existing paper records to a digital format. Because of this, they have a split responsibility between maintaining the existing servers, network and databases as well as creating interfaces for the front-end team’s applications. The Software Development team was recently formed to create new applications for both the staff and patients.

As shown in the Business Process graphic, the DCGH would like to update their processes to a digital format. The teams were tasked to analyse their existing documentation practices and adopt them to mobile apps or websites. Given the in house effort, limited requirements were given to the teams for them to work with. Instead, they were given an open ended request by hospital management to overhaul their paper processes.

# **2 Problem Analysis**

The DCGH software process issues stem from managing two separate teams with conflicting processes. The feature, capability driven Information Technology team adopted a waterfall approach, while the stakeholder feedback driven Software Development Team uses a hybrid agile/waterfall process. Each team selected their process based on their internal needs without the consideration of how the teams would collaborate together and relied on the project managers to facilitate integration events. This section will provide context for why these processes were selected as well as the challenges resulting from them.

Prior to the Software Development Team’s formation, DCGH formed the IT team to move their physical records to digital formats and build a back end infrastructure to network this data. The capabilities at this stage had a well defined scope with minimal deviation from schedule so the team naturally adopted a waterfall process. Since the systems were all new, there wasn’t a need to address bugs with updates and the delivery schedule was planned with major milestone deployments. These milestones are on the scale of one to two months coinciding with a major roll out of a system feature or capability. To exert as little impact on the greater hospital’s business processes, they focused on building a network of databases to store archived records with only an essential set of interfaces to digital forms for simple data entry into the system.

As for the Software Development team, the newly formed group works with mostly undefined requirements since the hospital’s stakeholders can vary from medical staff, administrative staff, or patients. The need to provide low cost prototype demonstrations to stakeholder groups is essential to define the front-end application’s requirements. This team functions on a hybrid process of agile and waterfall. At a high level, they have a waterfall-like structure with milestone delivery dates of application capabilities, but they’ve adopted agile processes of iterative development and prototype demonstrations to work with stakeholders to define their requirements.

Throughout development, the teams struggle to maintain consistent designs and interface definitions. The project managers plan the delivery schedule at the start of each project, but there is no standard collaboration between the development teams on designs or interfaces. Instead, the Software team will send interface descriptions to the IT team as they develop that part of the application. This results in the IT Team struggling to develop potentially large systems or major rework late in the schedule. In addition to the schedule delays, the IT team also has the challenge of bugs introduced from rushing development in order to meet delivery milestones.

Adding to challenges with late stage changes, the developers also encounter problems with misaligned priorities for what capabilities to develop first. Each project manager creates their schedule with the goal of meeting the final release date but does not consider that iterative testing will be needed during development to validate the design.

As the overall system grows in size and complexity, both teams are challenged with maintaining consistent bug free quality across all components. Currently, a formal method for testing system wide software and identifying bugs at release is not implemented. Each team conducts unit testing during development, but they do not have a formal method for testing the system after integration. This results in users finding bugs after release that then require rework and regression testing to address.

# **3 Proposed Software Process**

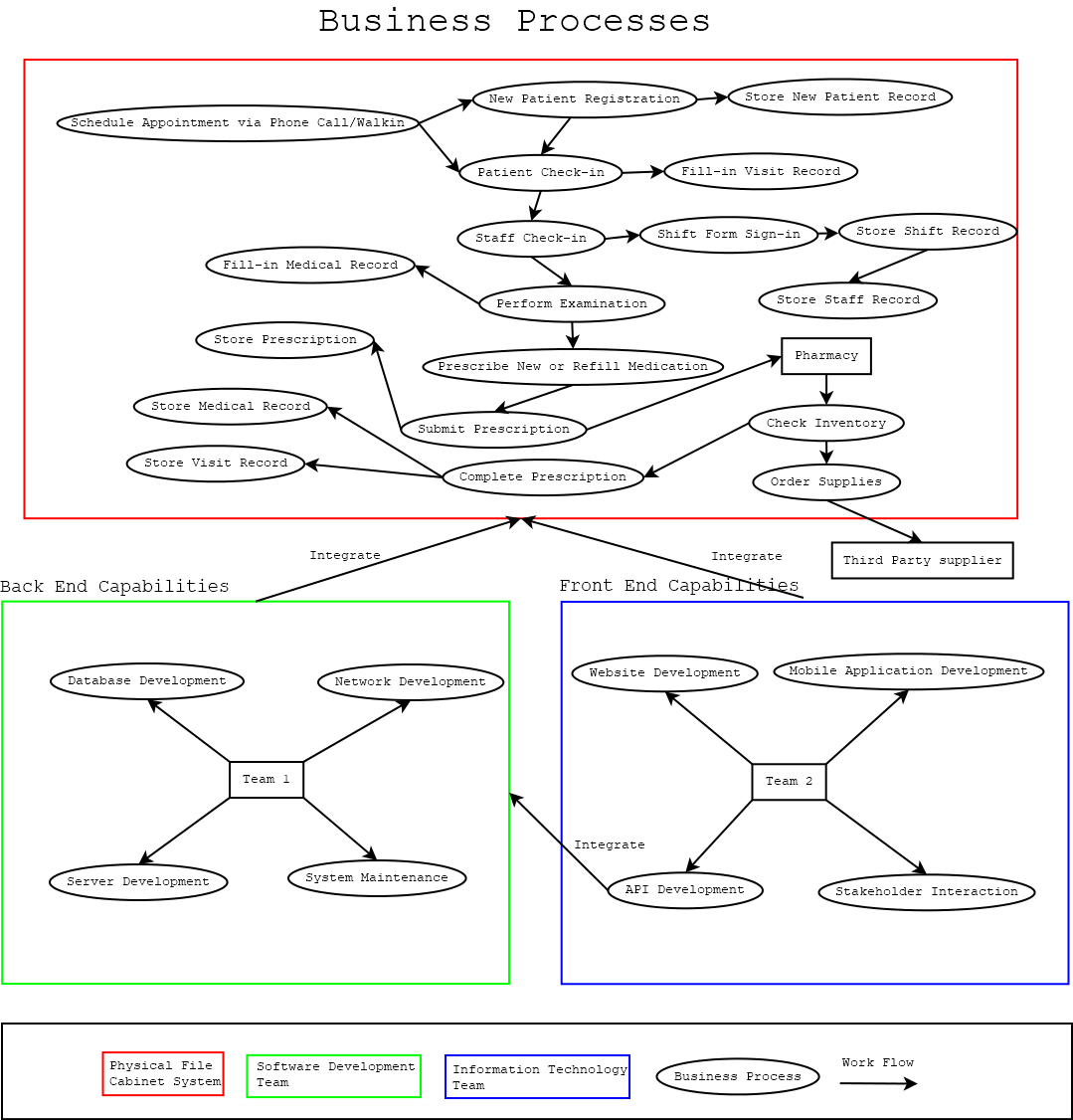
## **3.1 Business process Description**

DCGH has been updating the system in order to provide more efficient, more safe services to the increasing number of patients since the recent population growth in the surrounding area.

In the past, DCGH used to save medical records in a physical way, including the patient files, medical records, hospital course and so on. However, the physical records are way too cumbersome and time consuming. And it is tedious for the hospital staff to process with data for their patients. On top of this, physical records are difficult to preserve or transfer. The hospital has been upgrading the management system, from physical mode to more modern digital way.

All the information such as appointments, patient info, and prescriptions to a digital system. The medical system includes 1) front-end design (website and app) provides interactive services to hospital staff and patients 2) back-end system to store the old records in the database and new data records management. The DCGH management system can realize hospital online registration, electronic bills payment, online appointment etc. which can save costs, improves work efficiency, and reduces waiting time for patients in line.

The front-end system is supposed to provide interactive services between the hospital and patients, the hospital and doctors. The patients and doctors, as the users of the DCGH hospital, require clear UI design and easy-to-use personal information management. The system provides the user log-in, personal records searching, appointment making, prescription querying services and so on. The back-end system is supposed to store all the information safely and in a well-organized way. As the need for large scale data storage arises, the back-end system needs to support large processing. Apart from this, DCGH datasets include a large amount of personal information, such as contact information, medical history and other private information. Hance, any data leaking or corruption might put patients’ privacy on a big risk, which requires higher security from the database design.



**Figure 1: The current business process and new system to be integrated. The workflow in red borderline is the original physical file cabinet system, the green and blue frame are two parts of the new system. The front-end capabilities are the software development team’s responsibility, the back-end capabilities are the information technology team’s goal.**

## **3.2 Business problems analysis**

In the development process, the IT and software development team need to constantly collaborate with each other to ensure the system is on schedule and avoid any confusion or conflicts that may occur. The separation of front and back ends aims to improve efficiency, Clear Responsibility management, which makes the entire development process more flexible.

## 

## **3.3 Developer Team Roles**

The DCGH is organized into two teams of 10-15 developers each responsible for front and back end software development. The IT Team develops the back end consisting of databases, network infrastructures, and servers while the Software Development team is responsible for the front end including the hospital’s website, mobile applications, and Proprietary software.

To coordinate the teams better, a new role Technical Lead role will be introduced for each team that is responsible for communicating changes or future issues with the other team. Additionally, they serve as a technical advisor for the project manager and representative to the stakeholders. Any potential roadblocks can be brought up to the project manager by the technical lead so they can take the actions to organize a meeting or review. This change doesn’t remove responsibilities from the current organization, but instead covers a gap in technical leadership.

In the system design stage, the developers of the front-end and back-end architecture will analyze the system requirements together and agree to common interfaces or API. Having the developers own the design will establish that both teams understand what they're responsible for implementing. Following the design stage, the front-end and back-end personnel will jointly develop the common interface for their respective software components. In the project development stage, the front-end and back-end teams are separated by the responsibilities and tasks in order to achieve more efficient collaborative agile development. The back-end teams provide APIs, meanwhile the detailed descriptive documentation needs to be given. The front-end personnel perform page rendering and the front-end task is to send API requests and render the page after getting the data. In the project testing stage, before the API is completed, the front-end personnel will use a mock server for simulation testing, and the back-end personnel will use junit for API unit testing without waiting for each other team’s completed. Once the API is completed, the front-end and back-end docking tests can be done. Since not all interfaces can be defined in advance, some adjustments need to be made during the development process.

## **3.4 Collaborative development process**

*Stage 1* Evaluation: The product manager and the front-end and back-end teams conduct requirements reviews, make sure each team understands the responsibility, workload, workflow, and the schedule. A formal document is drafted and put into change management that details the planning conducted during Evaluation. The submission of the document outlines a clear plan that is mutually agreed to by the teams. During development this document can still be modified if requirements change, but the team will need to submit a change request to the documentation. As part of this request, the teams will need to review and agree to the proposed change before merging it into the formal document.

*Stage 2* Development preparation: The front-end and back-end teams discuss the parts that need to be jointly debugged together, and carry out informal agreement on the interface design. Create local interface simulation(including back-end template rendering), avoid a lot of online debugging which can be very inconvenient and time-consuming. Without local interface, the whole structure needs to be rebuilt and then synchronized to the server whenever the code updates.

*Stage 3* Interface definition: The front-end and back-end team, each draws up a detailed

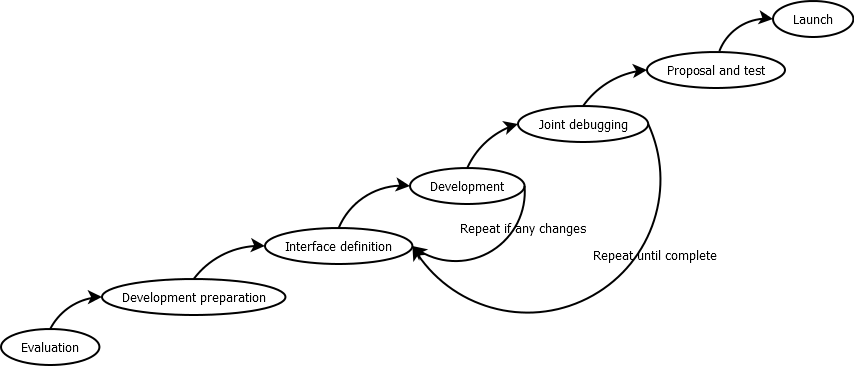
interface based on the previous informal agreement, and drafts the service interface design document, which needs to be confirmed after completion. Without interface documentation, development will need to pay more communication costs.

*Stage 4* Development: Both teams develop based on the interface. If anything needs to be added or deleted during the development process, repeat step 3. Each team is able to retain their existing development processes during this stage as long as the interface remains unchanged.

*Stage 5* Collaborative Integration: After the two teams have completed their independent work and start front-end joint debugging. If there are any conflicts and mistakes during the joint debugging process, repeat step 3 until the joint debugging is completed. Utilize mock data and test frameworks to run the project and do the interactive test and component unit test. Interactive tests make sure the interaction logic is reliable and interactive experience is ideal.

*Stage 6 Formal Quality Test*: Deliver test procedures for capability requirements verification to the quality assurance test team. If a defect is found, document and report the incident to the development team to conduct any needed regression work. and make changes on it until the product is free of bugs, and ready to be released.

*Stage 7* Release: Launch the system and continuously collect the feedback from users and make adjustments and updates regularly. Make sure the system can run stably.



**Figure 2: Collaborative development process**

## **3.5 Process Metrics Selection**

The system metrics selection criteria is based on five requirements that represent both development teams’ and stakeholder’s goals: identify areas of improvement, manage workloads, reduce overtime, reduce costs, and increase return on investment (ROI). The following sections describe these requirements and provide justification for how the proposed development process satisfies these needs.

### **3.5.1 Identify Areas of Improvement**

Since this project is an upgrade of the current business process and new integration to the old system, the improvement areas need to be identified and marked in the development process. The system metrics should provide details of improved areas before, during or after the development process to provide improvement data.

### **3.5.2 Manage Workloads**

Although the software development team and information technology team both have four components to be implemented, the API component in the front-end needs additional effort in order to establish communication between back-end capabilities and front-end capabilities. Therefore, the system metrics should provide workload management data before, during and after the development process to project management in order to help them make planning decisions.

### **3.5.3 Reduce Overtime**

Since both the software development team and information technology team are working simultaneously and collaboratively on different system capabilities, the overtime scenario is highly likely to appear during the development process. The system metrics should provide schedule related data during the development process to guide both teams and managers on the critical path to making adjustments and delivering the product on time.

**3.5.4 Reduce Costs**

The system metrics should provide cost estimation and actual cost information before, during and after the development process to help individuals, organizations and stakeholders to make financial changes and decisions.

### **3.5.5 Increase Return on Investment**

Unlike other business models, the purpose of Drexel County General Hospital Management System is to increase the efficiency and user experience in order to attract more customers. The organization’s goal is to expand the market and make more profit in order to scale company size. Therefore, the return on investment(ROI) data should be estimated and measured by the system metrics before, during and after the development process to inform stakeholders about financial facts and the effect of their business decisions.

## **3.6 Applied System Metrics**

As detailed in section 3.5, the system metrics will provide the team with the tools to efficiently execute tasks as well as align their performance with overall business goals. In order to track the effectiveness of the team, these metrics must also be applied to the Section 3.3 Collaborative Development Process: identify areas of improvement, manage workloads, reduce overtime, reduce costs, and increase return on investment (ROI).

### **3.6.1 Evaluation**

Starting with the Evaluation stage, the product manager and the front-end and back-end teams conduct requirements reviews, make sure each team understands the responsibility, workload, workflow and the schedule. Therefore, the training is the best system metric for this stage because it provides visibility into the effectiveness of the organization’s training program in meeting the skill requirements(1).

First, the project manager utilizes training metric data along with requirement reviews to drive these tasks of leading the development teams: dispatch workloads, define responsibilities, design the workflow, and plan team schedules. Considerations for assigning these tasks to either the information technology or software development team will need to account for each individual's training performance and skill set. Secondly, the training metric data shows skill requirements for all system capabilities, so it can be used to identify areas of improvement by cross comparing those areas with a team or individual’s skill set. Thirdly, the training metric data can also help the project manager to assign appropriate workloads to team members that compliment their strengths and fit the skill requirements needed to complete a given task.

Fourthly, the training metric data can help the project manager, team or developer reduce overtime by making personnel adjustments based on skill sets and task’s skill requirements. Last, the training metric data shows the cost of training programs. It allows project managers to make estimation and calculate the cost of using both teams and all individuals, as well as the financial estimation about the workload, workflow, schedule and requirement, which is useful in gathering information and making decisions about reducing costs and increasing return on investment(ROI).

### **3.6.2 Development Preparation**

The Level 2 Development Preparation stage is for front-end and back-end teams to discuss the parts that need to be jointly debugged together, and carry out informal agreement on the interface design. The effort is the system metric to be used in this stage because it provides visibility into the contribution that staffing has on project costs, schedule adherence, and product quality(1). The effort metric data helps the software development team and information technology team make estimation and calculate actual effort about time and resources be used in debugging and interface design. First of all, it can be used to identify areas of improvement. If the effort metric shows certain interface design or debugging process’s effort is much higher than the others, then improvement needs to be made. Secondly, the effort metric data can be used to manage workloads. If some workloads’s estimate or actual effort is large, then the project manager should consider adding more people or extending the schedule to implement them. Thirdly, it can be used to reduce overtime. By using effort metric data, the project manager can make more precise schedules based on system capabilities’ current or estimated effort. Lastly, with the effort metric data and other information, the stakeholders can calculate the current and future cost of interface design and joint debugging, including the cost of time, money and other resources, to help them make financial changes and decisions in order to reduce costs and increase return on investment(ROI).

### **3.6.3 Interface Definition**

In the Interface Definition stage, the software development team and information technology team each design the detailed interface based on the previous informal agreement, and drafts the service interface design document, which needs to be confirmed after completion. The cost is the system metric to be used because it provides tracking of actual costs against estimated costs and predicts future project costs(1). The cost metric data is a complement to the interface definition stage. Since the interface definition is the start of the development process, it is a complete development process and represents many characteristics of the development process. First of all, the cost metric data can be used to identify areas of improvement. If a certain interface definition area’s cost is higher than the others, then improvement needs to be made, as well as the similar parts in the following development process. Secondly, it can help the project manager dispatch workloads by assigning skillful and experienced people to handle costly parts. Thirdly, the cost metric data can help reduce overtime. If a certain part is very costly, then it must take many resources. The project manager should consider extending the schedule or making personnel changes. Lastly, the cost metric data can be used to calculate overall cost of the project by multiplying the component numbers. Since the interface definition is the first process, the cost metric can be used to help reduce costs and increase return on investment(ROI) by making modifications on costly parts in this stage, as well as the related capability components in the following development process.

### **3.6.4 Development**

The Level 4 Development stage is when both teams implement the on the designed software components and interfaces. If anything needs to be added or deleted during the development process, repeat Interface Definition in the Development stage so both teams agree to the change. The progress metric is the system metric to be used here because it provides information on how well the project is performing with respect to its schedule commitments(1). Since the Development stage conducts actual implementation, the progress metric is a good indicator not only for system measurement in this stage, but also for the Development and Interface Definition cycle as well as the whole project.

The progress metric data can be used to identify areas of improvement. If a certain part’s progress performance is not as expected, then improvement needs to be made. In the event a certain part’s progress performance is not as expected, the manager should consider making changes about developers or workload distribution. This will alleviate any disproportionately assigned tasks to a given developer.

The progress metric data will also help reduce overtime. If a certain part’s progress performance is not as expected, the progress metric data allows the project manager to make plan changes or personnel changes on time to avoid overtime. By combining progress metric data and other data in this stage, it is easy to calculate the cost of this stage and overall costs of the whole project. If any part’s progress metric data is not expected, this information allows stakeholders to react immediately by making financial changes and decisions to reduce costs and increase return on investment(ROI), for both current stage and overall development process process.

### **3.6.5 Collaborative Integration**

The Level 5 Collaborative Integration phase begins when the two teams have completed their independent work and start front-end integration. If there are any conflicts and mistakes during the Collaborative Integration process, return to Interface definition phase and then development stage to implement the change until the interfaces are integrated. Regarding metrics, Stability should be used in this stage because it provides visibility into the magnitude and impact of requirements changes, insight into the completeness and stability of the requirements and into the capability of the staff to complete the project within the current budget and schedule, as well as insight into the effectiveness and quality of the defined process(1).

Since level 3 Interface Definition stage and Collaborative Integration phase is a development process, the stability metric is not only a good indicator for this stage, but also for the whole development process. The stability metric data can help identify areas of improvement. If one or more stability results shows the requirement, size, or process performance is not as expected, the improvement needs to be made while also identifying the target area.

Additionally, the Stability metric can help manage workloads. If one development process’s certain stability metric data is as expected, then the project manager should transfer some developers from that area to work on other workloads which belong to a bad stability performance area. By improving the stability performance based on data in this stage, the future development process’s performance will be improved which can save cost and ultimately reduce overtime. Aggregating the improvements over time, the stability metric data can help mitigate inefficiencies and increase return on investment(ROI) by allowing the project team to achieve expected stability performance which reduces the maintenance cost.

### **3.6.6 Formal Quality Test**

The Level 6 Formal Quality Test stage proposes the completed requirement for testing purpose. If a problem is found, they will notify the development team in time and make changes on it until the product is free of bugs, and ready to be released. The quality metric is the system metric to be used in this stage because it provides estimation of product quality as the software is verified to satisfy the stakeholder requirements. Fulfilling these requirement confirm the project processes, insight into the quality of the product and processes and the effectiveness of testing, insight into the quality of the intermediate and final products and into the peer review and development processes, as well as insight into the cause of defects and the effect of defect prevention activities on defect insertion rates(1).

Since the purpose of this stage is to validate requirements and report defects, the quality metric aligns with the output results from this stage. As defects are found in formal testing, the metric will reflect the effectiveness of each team at delivering functional software. Reports such as defect density will provide project management with a threshold indicator that the development processes need to be reviewed to mitigate bugs found in release builds.

The quality metric data can also help manage workloads by assigning a team or individual to address components with excessive reported defects. The quality metric data will also reduce overtime spent on addressing multiple issues with a single component by giving management an indicator that the team should revisit a capability to find a root cause.

Lastly, the quality metric data can help reduce costs and increase ROI. Since this data can help developers locate and find defects in the product, the stakeholders can make financial changes or decisions based on the quality performance. On the other hand, the defect prevention data in the quality metric help product owners or operators reduce the cost of maintenance.

### **3.6.7 Release**

The Level 7 Release stage is the deployment of the feature capability. Feedback from users is monitored by the project managers to ensure the system is stable. For this phase, the computer resource utilization metric will be used because it provides information on how well the project is performing with respect to its computer resource utilization goals or requirements(1).

The computer resource utilization metric is a good indicator for this stage as the team monitors the performance impact and maintenance of the system. Maintainers can use this data to identify the potential areas of low performance components or incompatibilities with hardware as the system continues to grow over its lifecycle. This is important for software deployed in hospitals as they must support a wide range of hardware including older legacy equipment.

# **4. References**

1. Baumert. John, and McWhinney. Mark, "Software Measures and the Capability Maturity Model," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Report CMU/SEI-92-TR-025 , 1992.
2. Cockburn A. Agile software development[M]. 2nd ed. New York: Pearson Education, Inc, 2007.
3. Software & Systems Engineering Standards Committee (C/S2ESC). 1648 WG-RP for establishing and managing software development efforts using agile methods[EB/OL]. [2016-09-18].