# Information System Analysis, Design, & Implementation – Mamma Mia

Steven Duchene

MIS 350 – Festus Onyegbula

Colorado State University – Global Campus

December 1, 2019

Information System Analysis, Design, & Implementation – Mamma Mia

## Phase 1 – Planning & User Requirements

### Company and the System's Environment

Mamma Mia is a food product company that produces and distributes dry pasta products nationwide. Currently, Mamma Mia is using a relatively modern and reasonably complete, in-house developed, information system; an exception being the Purchasing and Accounts Payable systems. Mamma Mia is currently utilizing client-server-server architecture based on large HP-UX servers running an Oracle database. All company servers are hosted at the company's Denver headquarters to which all other locations connect to the server farm using redundant communication links. The company is satisfied with their uptime and down time statistics. At this time is happy with their integrated system.

### Current System Analysis

Accurate requirement definition with regards to a systems design can drastically reduce the number of extensive and costly design errors. In order to accurately determine the requirements of the purchasing systems currently in use at Mamma Mia, a detailed requirements determination study was completed. Data collection techniques utilized in this study included the conducting of face-to-face interviews, group interviews, and direct user observation. Additionally, an in-depth analysis was conducted with regards to current system procedures and organizational documents. Below is a description of the current purchasing system in use at Mamma Mia.

Mamma Mia currently utilizes an aged procurement software system that has not been updated since its implementation 20 years ago. While the procurement system originally implemented was effective, it did not meet the completed needs of the organization and a large

amount of data was still compiled in to Excel spreadsheets. Furthermore, the procurement system does not provide an adequate means of inventory management and requires that data from a paper-based inventory system be manually entered; one that requires a large time commitment. In its current state, the purchasing and procurement system has reached its peak capabilities and is no longer scalable to meet the needs of Mamma Mia's continual growth in the food product marketplace.

**Detailed Requirements for the New System**

At the request of the organization, the new system must be a stand-alone, independent system with the intent to have the system incorporated into the other information systems, already in use at Mamma Mia, at a later date. As such, this system must be designed so that it will integrate into the current database, Oracle, thus the designed software must be compatible with the Oracle technology. Firstly, the new procurement system should provide a digital procurement solution so as to reduce or eliminate maverick spending. Secondly, the system must accommodate a bidding process for larger orders. In doing so, Mamma Mia will increase the competition for their orders which will reduce prices. Thirdly, the procurement system must include features such as inventory management, inventory planning technologies, work order management, and automated ordering and purchasing. By providing real-time updates on stock levels, the software an accurately predict inventory requirements based on sales trends. Finally, the system must be able to centralize orders across multiple locations and departments; reducing labor costs associated with procurement. D

While the aforementioned requirements deal primarily with the functionality of the upgraded procurement system, special attention must also be paid to the technological requirements of a new system. As stated in the previous paragraph, the upgraded system must be

compatible with the Oracle system currently in use at Mamma Mia. Other important factors that are critical to the new procurement system are: maintainability, scalability, reliability, and user experience and support. In order to provide a greater insight with regards to the technological requirements of the system, some explanation is required. Maintainability references the ease at which the software solution can be maintained in order to correct errors, fix bugs, and adjust/add features to the system. Scalability is a reference to the systems ability to grow, handling increased system demands without losing system efficiency (Odhiambo, 2019).

**Feasibility Analysis**

Investments in information technology are inherently risky due to the unknown economic impacts, technological complexity, and other challenges (Dewan et al, 2007). An important question to ask is this, "What is the impact of IT risk on the required rate of return on IT investment and on the productivity and market value" (2007). With the design and implementation of an upgraded procurement system, Mamma Mia can see organizational benefits such as a reduction in the number of steps in the inventory control process, obtain real-time inventory updates, schedule automated reorder points on products, and also the capability to integrate procurement into Enterprise Resource Planning software.

When assessing feasibility, factors that must be considered include economic, technical, operational, time, legal, and political factors (Valacich & George, 2018). A feasibility study was completed in which the above factors were assessed. Costs of system implementation will include server software, additional servers, development costs, and training costs; including financial and time resource costs. Benefits of system implementation are presented below.

- Economic/Financial: Tangible economic benefits include cost reduction as a result of reduced errors, higher accuracy, and accurate forecasting. Intangible benefits include

higher operational efficiency and associated cost savings. Research has shown that for every $1 of IT investment made, an increase of $10-$15 is observed in market valuation (Dewan et al, 2007).

- Technical: Greater efficiency in data processing and storage.
- Operational/Time: Real-time information and analytics, improved processing efficiency, increased employee morale, improved accuracy, and increased organizational planning.
- Political: Better usage of resources (green revolution).

**Project Structure using SCRUM**

Scrum project management is an iterative process in which risk is controlled and optimize the predictability of the project (Lei et al, 2017). Key factors of the SCRUM process to be used include the following:

- Transparency: The process activities will be visible to all stakeholders
- Inspection: End product users will be involved in the testing and inspection of project deliverables.
- Adaption: Test group feedback can and will adjust the project process (Lei et al, 2017).

**Project Structure.**

1. SCRUM Team Assignments
    a. SCRUM Master: Manager of team who ensure the team adheres to process, rules, and theory of the methodology.
    b. Product Owner: Representative of the stakeholder and guide for product development.

   c. SCRUM Team Members: self-organizing and cross functional, comprised of individuals with diverse skillsets and a high drive for success (Srivastava and Jain, 2018).

2. Project Activities
   a. Backlog Organization
   b. Planning Sessions
   c. Sprint
   d. SCRUM Meetings
   e. Sprint Review
   f. Sprint Retrospective

**Project Initialization.**

The SCRUM methodology is the series of events listed above. Project initialization will begin with the determination and assignment of SCRUM team roles. Once roles have been determined, the SCRUM processes will begin with the backlog organization stage. During this stage, project requirements are determined in order to meet the session and overall project deliverables. Requirements will be sorted in order of priority and all subsequent events will take place daily.

At this stage of the process, the SCRUM team will meet with the stakeholder to determine system requirements, establish work standards, and additional procedures.

**Intermediate Deliverables**

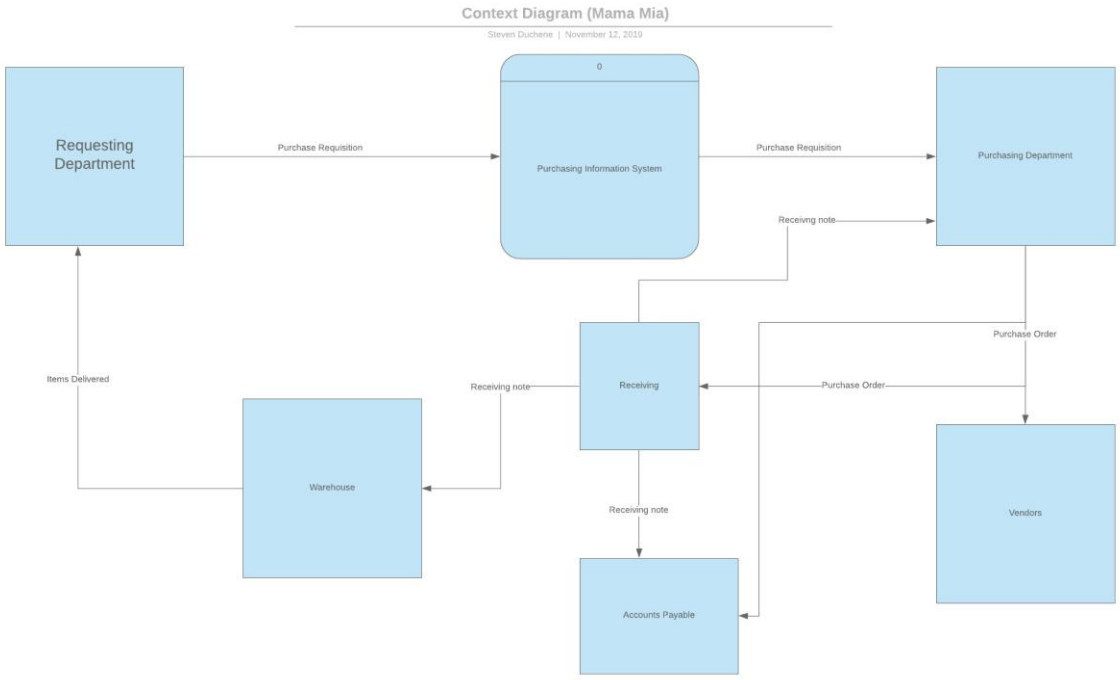
Intermediate deliverables for this stage of the process will include the following, tangible products:

- Baseline Project Plan

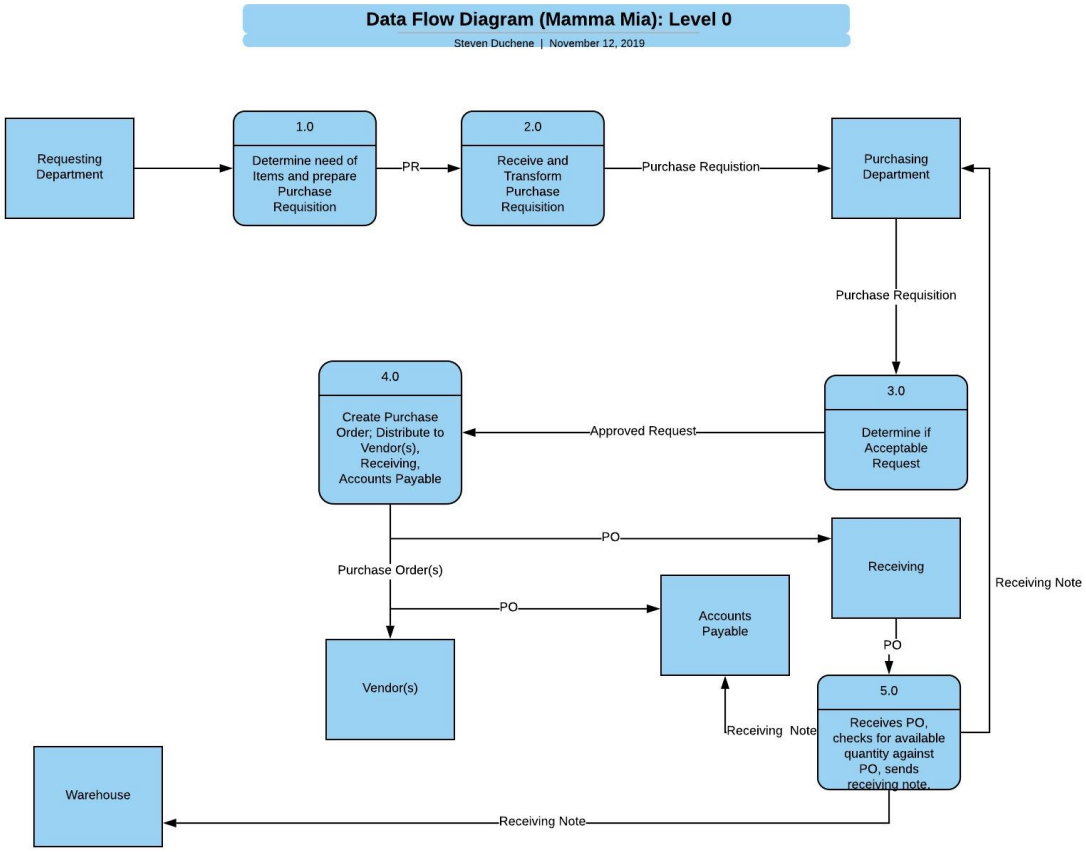
- Project Scope Statement
- Project Feasibility Report

## Mamma Mia Data Flow Analysis

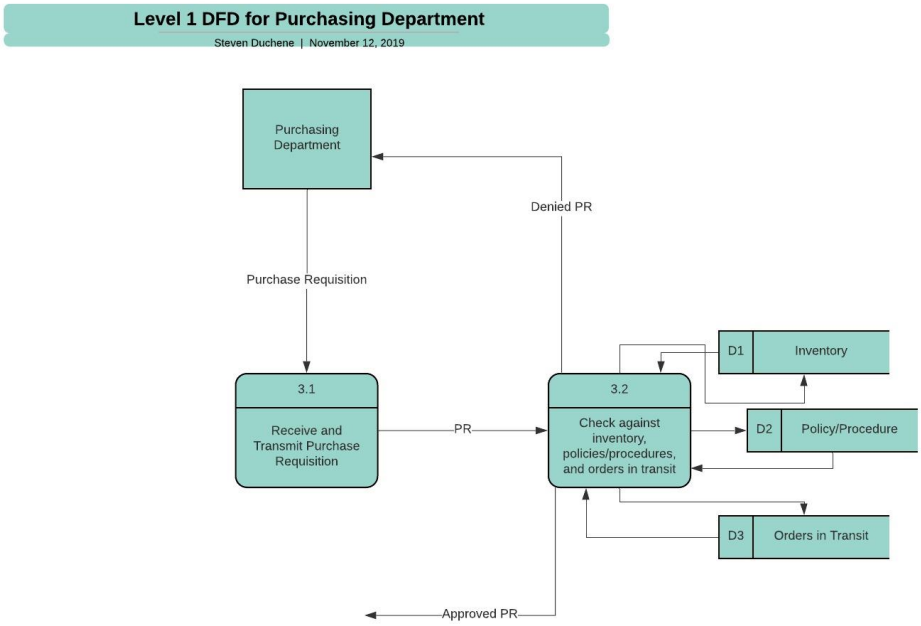
### Context Diagram: Mamma Mia



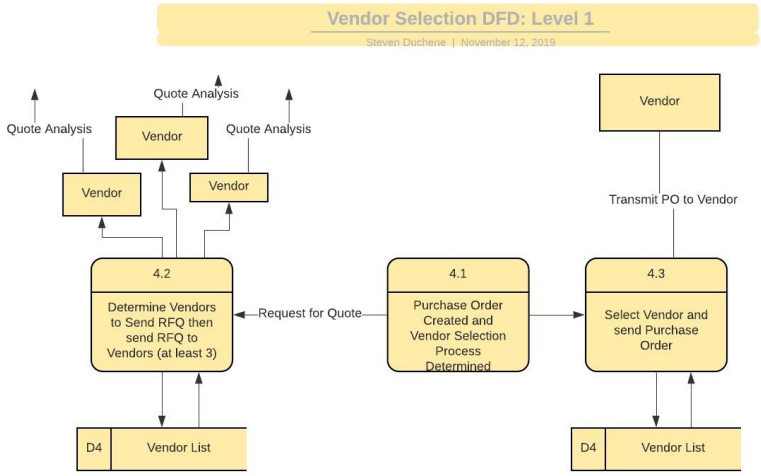
## Level 0 DFD: Mamma Mia



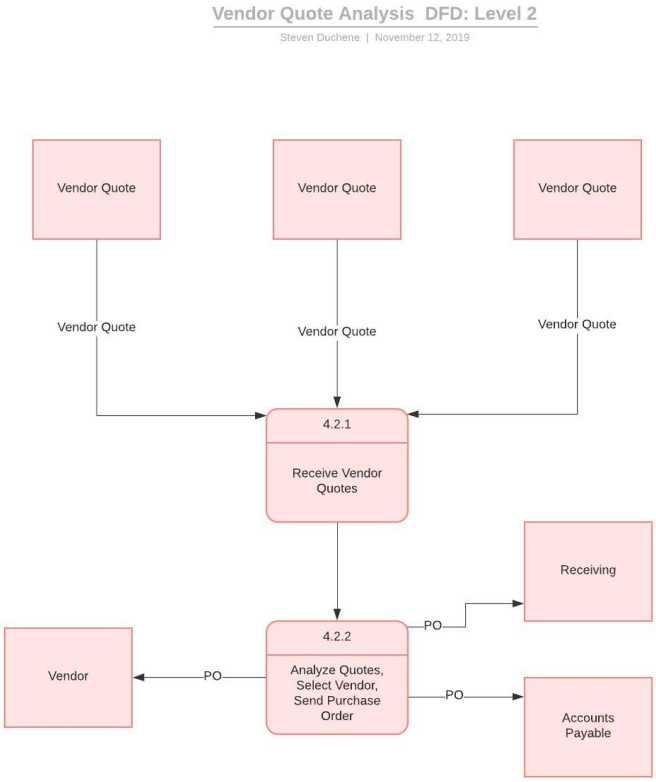
## Level 1 DFD: Mamma Mia Purchasing Department



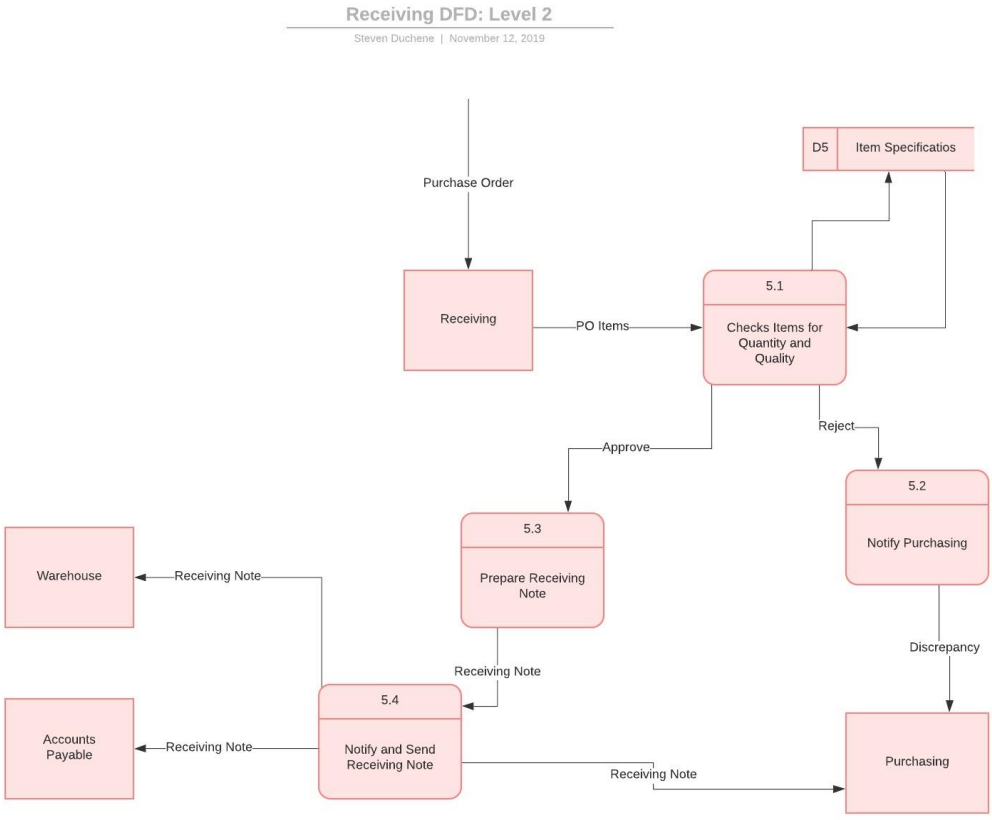
## Level 1 DFD: Mamma Mia Vendor Selection



## Level 2 DFD: Mamma Mia Vendor Quote Analysis



**Level 2 DFD: Mamma Mia Receiving**

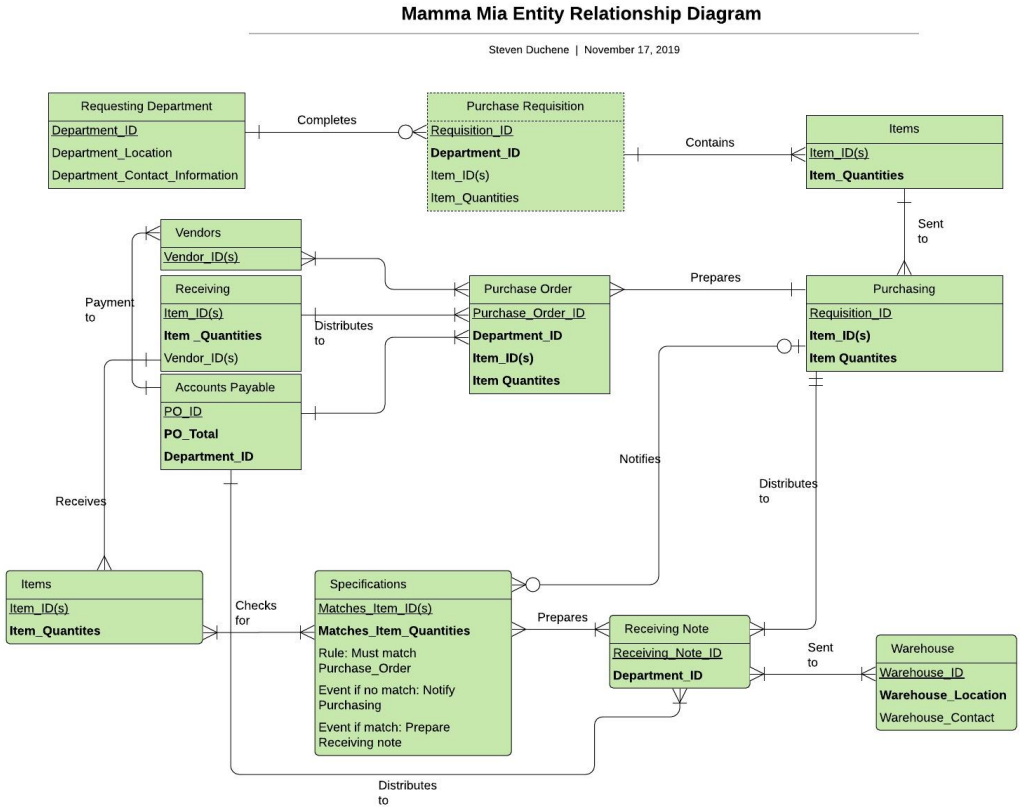



## Entity Relationship Diagram & Database Design: Mamma Mia

### Entity Relationship Diagrams

The entity-relationship modeling process is the development of a comprehensive, logical representation of the data for an organization (Valacich & George, 2018). Through the conduction of the data flow analysis, a sort of reverse engineering, we have uncovered program logic and observed the functions required for the new system. Furthermore, we have determined the processes in which data is captured, manipulated, stored, and distributed within the Mamma Mia system. At this point, we begin the process of developing the Entity Relationship diagram for the Mamma Mia System. Before creating a dependable, working system, it is necessary to

determine the entities, relationships, and attributes of the data system of study. Furthermore, it is necessary to construct a dynamic, informational view of the system (Chen & Lu, 1997). We continue our study and analysis of the Mamma Mia purchasing system with the development of the Entity Relationship Diagram.




## Physical Database Design

At this point in the system development life cycle, it becomes necessary to take the following steps when developing the physical database design:

1. Develop logical data models for each known interface using the principles of normalization.

2. View Integration: Combine normalized data requirements into one logical database model.
3. Translation of the Entity Relationship Diagram into normalized data requirements.
4. Comparison of the logical database design with the translated ERD resulting in the final logical database model for the application (Valacich & George, 2018).

After accounting for every data element possible in the system, the acquired data can then be translated into the architecture that will ultimately result in an operating system. One major purpose of this phase of the SDLC process is reduce the amount of space and energy required by the proposed database. The high demand of organizations to efficiently store, access, and manage their data is not only an organizational issue but an environmental and political issue as well (Roukh et al, 2017). This section will present the physical database design model and all associated information.

1. Entities Denormalized
    a. PURCHASE REQUISITION(Requisition_ID, **Department_ID, Item_ID(s), Item_quantities**)
    b. PURCHASING(Requisition_ID, **Department_ID**)
    c. PURCHASE ORDER(Purchase_Order_ID, **Requisition_ID, Department_ID**)
    d. VENDOR(Vendor_ID)
    e. RECEIVING(Purchase_Order_ID, **Item_ID(s), Item_Quantities**)
    f. ACCOUNTS PAYABLES(Purchase_Order_ID, **PO_Total, Department_ID**)
    g. SPECIFICATIONS(Matches_Item_ID(s), **Matches_Item_Quantitites**)
    h. RECEIVING NOTE(Receiving_Note_ID, **Department_ID**)

i. WAREHOUSE(Warehouse_ID, **Receiving_Note_ID, Warehouse Location,** Warehouse_Contact)

## Purchasing Implementation with DevOps – Mamma Mia

The prior sections have described the analysis and design for the Mamma Mia Purchasing Information System. The focus of this section will be multifold including the detailed descriptions of testing, user training, and system implementation utilizing the practice of DevOps. In order accurately describe the final processes of this project, one must first describe the practice, or set of practices, that is DevOps. DevOps, also known as development operations, is the set of practices that serve to automate the progression between software development and information technology teams; facilitating the building, testing, and release of software faster and with greater reliability (Atlassian, n.d.).

As the quickly emerging and preferred platform for software development and implementation, DevOps is pushing the barriers with regards to new approaches, tools, and artifacts in the Systems Development Life Cycle. How can it benefit Mamma Mia? As a method that fosters collaboration, faster releases, error resolution, and positive culture; it is the ideal process for the implantation of the purchasing information system for the Mamma Mia Company. The following sections will detail the test plan, training plan, and further describe the DevOps process including details such as why DevOps, its associated costs, the financial benefits, and pertinent other benefits.

### Test Plan

In order to ensure that a functional system is being developed, system testing will occur throughout the process. As in the Systems Development Life Cycle, system testing is something that is planned from the very beginning of the SDLC process. While physical testing may not

occur until later development stages, the master testing plan will be developed during the analysis phase. The Master Testing Plan, while not developed for the purposes of this proposal should be modeled after the following (Valacich & George, 2018):

| | |
|---|---|
| 1. Introduction<br>a. Description of system to be tested<br>b. Objectives of the test plan<br>c. Method of testing<br>d. Supporting documents<br>2. Overall Plan<br>a. Milestones, schedules, and locations<br>b. Test materials<br>i. Test plans<br>ii. Test cases<br>iii. Test scenarios<br>iv. Test log<br>3. Testing Requirements<br>a. Hardware<br>b. Software<br>c. Personnel | 4. Procedure Control<br>a. Test initiation<br>b. Test execution<br>c. Test failure<br>d. Access/change control<br>e. Document control<br>5. Test-Specific or Component-Specific Test Plans<br>a. Objectives<br>b. Software description<br>c. Method<br>d. Milestones, schedule, progression, and locations<br>e. Requirements<br>f. Criteria for passing tests<br>g. Resulting test materials<br>h. Execution control<br>i. Attachments |

.

Throughout the testing phase, several different and specific types of tests will need to be conducted. As we are utilizing the SCRUM method of development, testing will occur as a part of the SCRUM cycle.

## Inspection

The first type of testing that will occur is the inspection phase of testing. During this phase, developers will examine the written code in search of predictable, language-specific errors in the code (Valacich & George, 2018). Inspection testing will be conducted by developers participated in the writing of the applicable codes. Furthermore, the code will be examined against known errors for the programming language.

## Functionality Testing

The second type of testing that will occur will be a basic functionality testing of the system. In its most basic form, the functionality testing ensures that the system functions as expected. While not all possibilities of clicks and characters will not be tested, the goal is to detect items and codes that do not function as expected.

## Code Review

At this point in the testing process, the third type of testing will occur; code review. As the saying goes, “Since our own eyes deceive us, it is best to have another set of eyes review.” Code review is conducted in the form of a peer review where other developers will review the code and suggest corrections, if necessary. While this stage may appear similar to the aforementioned inspection phase, it is different in the sense that developers that did not participate in the authoring of the code will be conducting the code review.

## Static Code Analysis

The forth phase is a more critical, yet automated form of code review termed static code analysis. During static code analysis, the analysis tool will look for many weaknesses in the code including security vulnerabilities and concurrency issues (Issacs, 2019). It is pertinent that the static analysis tools be configured to run automatically and enforce the correct coding standards applicable to the coding language. The next phases to occur begin the most critical phase of software testing: unit testing, integration testing, and system testing.

## Unit, Integration, and System Testing

At this phase of the testing process, developers will begin by writing unit tests in which individual modules will be tested in a further attempt to identify any errors occurring in the code that were not discovered by the prior testing steps. The tests will test across a range of situations and test a range of valid and invalid user inputs. As is the nature of the SCRUM method, unit tests will be conducted after any point where a change has been made to the underlying code.

Integration testing is where the bread meets the butter, per say, and a point in the testing process where the modules are combined for the purpose of testing nearly the entire system. This process should be conducted in a top-down approach where modules are combined and tested

together in a piecemeal approach. As tests succeed, additional modules will be added and tested accordingly (Valacich & George, 2018). This procedure will continue until the entire program has been tested successfully. While this appears to be a test of the entire system, this does not occur until the system testing phase.

In the prior phase, the modules that comprise individual programs were tested in their entirety to ensure functionality. During the system testing phase, these individual programs are subsequently combined and tested to ensure further functionality. Similar to the prior phase, the programs are combined utilizing a top-down approach and tested incrementally. The final phase of the testing cycle brings us to the single-user performance testing of the system. Occurring on the front-end of the system, this phase of testing is conducted to ensure that the software is responsive to the front-end user and working as expected. During this phase of testing, characteristics such as speed and system response are evaluated. Furthermore, the systems capability to communicate with other organizational platforms will also be tested. By conducting end user testing, developers will be able to identify user issues with the system and make corrections to the applicable code.

**Training Plan**

Training on the new system will occur in phases and will include the following types of end-user training events:

- Video Instruction
- Self-paced Computer-Based Trainings
- In-Person Instructor Led Trainings
- Hands-on Training
- Continuing Education and Remedial Training Sessions

It will be important that at every level of the organization, training sessions and material will be tailored to individual job requirements at each level. Furthermore, it is pertinent that training materials be developed so that learners of different styles will be able to hold on to the information they are given. Finally, the training aspect of the software development must include feedback, question and answer sessions, and metrics to measure the trainees grasp of the material

**DevOps Also Known as Development Operations**

As stated in the introduction to this final section of the report, DevOps, also known as development operations, is the set of practices that serve to automate the progression between software development and information technology teams; facilitating the building, testing, and release of software faster and with greater reliability (Atlassian, n.d.). The need for DevOps came from the disconnect that was being seen in separate systems development and operations teams. Systems engineers were in need of a better set of tools at their disposal to detect and measure problems in the systems they were developing (Roche, 2013). The questions that remain as to the use of DevOps include its benefits, costs, and financial benefits. The answer to these questions will serve to answer the following: Why does Mamma Mia need DevOps? The following will serve to answer these questions.

## Financial Benefits

Some of the biggest benefits from utilizing DevOps in systems development are the methods ability to unequivocally communicate describe the financial costs of systems development. As in any organization, financial statistics are critical to the decision making process. While the ultimate goal of DevOps is to release software faster and with higher quality, it is also important that investors see a return on their investment. While some measures of the benefits of DevOps are not easily translated into financial returns of investments, there are

several that can be measured. These include lower IT costs, increased revenue and growth, less down time, and greater service availability (FEI, 2016).

## The Cost of DevOps

The transition to DevOps, while necessary, is associated with additional costs with regards to its implantation. It must be clear to the organization the associated financial and labor costs that can be incurred. According to a survey conducted by KMS Technology, the costs associated with the transition to the DevOps model can range anywhere from $100,000 to $500,000; with a substantial portion of the investment going towards DevOps tools and technologies (Elizabeth, 2017).

## Why Does Mamma Mia Need DevOps?

As the quickly emerging and preferred platform for software development and implementation, DevOps is pushing the barriers with regards to new approaches, tools, and artifacts in the Systems Development Life Cycle. How can it benefit Mamma Mia? As a method that fosters collaboration, faster releases, error resolution, and positive culture; it is the ideal process for the implantation of the purchasing information system for the Mamma Mia Company. DevOps is becoming increasingly necessary to keep up with the growing demands of todays marketplace. By using DevOps, organizations are able to deliver system updates and features at a much quicker pace; maximizing the organizations competitiveness in the marketplace.

The decision to switch to the DevOps approach to software development does not come easily. It is a decision that must be made with a complete understanding of the associated costs and challenges that come with the change in methodologies. DevOps can improve an organizations ability to develop and implement new software. Furthermore, DevOps can result in

a happier organizational culture, greater employee engagement, and faster development of more stable operating systems.

**Implementation Strategy and Plan Utilizing DevOps**

Once the target system has been developed and tested, the time comes to implement the system into the organization; also known as installation. During this phase in the process, the system is officially installed into the organization and placed in to a working status. In order to install the system with the least affects seen by the organization, a phased installation of the system will occur. A phased installation is an incremental approach to system installation where changes from the old IS to the new IS occurs gradually until the new system is the only operating system in use (Valacich & George, 2018). It should be noted that when using the phased installation approach, the new and old systems must still be able to communicate. This will likely require the development of bridge programs to ensure adequate communication between the old and the new.

Installation of software components will occur during the off-hours of the organization as to ensure a lapse in systems operation does not occur. The installation schedule will be announced prior to it occurring; allowing users to prepare for any changes that may occur. Additional procedures will be necessary to ensure that data is not lost or destroyed should an emergency occur. To ensure that system implementation is successful, user involvement must occur, users must be invested in the new system, committed to the project, and be willing to change their behaviors. This can all be achieved by utilizing open communication and user involvement throughout the entire development of the system.

## Conclusion

As demonstrated by this report, the process of analyzing, designing, and implementing a new system is no easy task. Although there are many ways of achieving success in a project of this scope and size, the organization must be careful to follow the aforementioned steps and procedures outlined in this report. By adopting a SDLC and DevOps approach to system design and implementation, the Mamma Mia Company can ensure the overall success of the project. As the organization matures, so does its ability to adapt and overcome market challenges. By adopting the recommendations outlined in this report, Mamma Mia will put themselves on the road to organizational success.

# References

Atlassian. (n.d.). What is DevOps? Retrieved from https://www.atlassian.com/devops.

Chen, K.-Y., & Lu, S.-S. (1997). A Petri-net and entity-relationship diagram based object-oriented design method for manufacturing systems control. *International Journal of Computer Integrated Manufacturing*, *10*(1-4), 17–28. doi: 10.1080/095119297131156

Dewan, S., Shi, C., & Gurbaxani, V. (2007). Investigating the Risk--Return Relationship of Information Technology Investment: Firm-Level Empirical Analysis. Management Science, 53(12), 1829–1842. https://doi.org/10.1287/mnsc.1070.0739

Elizabeth, J. (2017, October 23). What is the true cost of a DevOps adoption? Retrieved from https://jaxenter.com/true-cost-devops-adoption-138287.html.

FEI. (2016, July 20). A Deeper Look at the Financial Return on Investment for DevOps. Retrieved from https://daily.financialexecutives.org/deeper-look-financial-return-investment-devops/.

Isaacs, M. (2019, November 1). 5 key software testing steps every engineer should perform. Retrieved from https://techbeacon.com/app-dev-testing/5-key-software-testing-steps-every-engineer-should-perform.

Khoo, B., PhD. (2011). A process-based analysis of object-oriented system analysis and design. International Journal of Management and Information Systems, 15(1), 123-130. Retrieved from https://csuglobal.idm.oclc.org/login?url=https://search-proquest-com.csuglobal.idm.oclc.org/docview/853756965?accountid=38569

Lei, H., Ganjeizadeh, F., Jayachandran, P., & Ozcan, P. (2017). A statistical analysis of the effects of Scrum and Kanban on software development projects. Robotics and Computer Integrated Manufacturing, 43, 59–67. https://doi.org/10.1016/j.rcim.2015.12.001

Lucid Chart. (n.d.). Online Diagram Software & Visual Solution. Retrieved November 17, 2019, from https://www.lucidchart.com/pages/?noHomepageRedirect=true.

Odhiambo, D. (2019, May 9). System Design in Software Development. Retrieved from https://medium.com/the-andela-way/system-design-in-software-development-f360ce6fcbb9.

Online Diagram Software & Visual Solution. (n.d.). Retrieved from https://www.lucidchart.com/pages/?noHomepageRedirect=true.

Roche, J. (2013). Adopting DevOps Practices in Quality Assurance. Association for Computing Machinery. Communications of the ACM. Retrieved from http://search.proquest.com/docview/2049933990/

Roukh, A., Bellatreche, L., Bouarar, S., & Boukorca, A. (2017). Eco-Physic: Eco-Physical design initiative for very large databases. *Information Systems*, *68*, 44–63. doi: 10.1016/j.is.2017.01.003

Srivastava, P., & Jain, S. (2017). A leadership framework for distributed self-organized scrum teams. *Team Performance Management: An International Journal*, *23*(5/6), 293–314. doi: 10.1108/tpm-06-2016-0033

Valacich, J. S., & George, J. F. (2018). Modern systems analysis and design. Singapore: Pearson Education South Asia Pte Ltd.