Fall 2010

Improving Requirements Elicitation By Leveraging the Discipline of Screenwriting

Albert Gardella
Regis University

IMPROVING REQUIREMENTS ELICITATION BY LEVERAGING THE DISCIPLINE OF SCREENWRITING

A THESIS

SUBMITTED ON THE 4TH OF SEPTEMBER, 2010

TO THE DEPARTMENT OF INFORMATION SYSTEMS, INFORMATION TECHNOLOGY,

COMPUTER SCIENCE

OF THE SCHOOL OF COMPUTER & INFORMATION SCIENCES

OF REGIS UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF MASTER OF SCIENCE IN

DATABASE TECHNOLOGY

BY

Albert Gardella

APPROVALS

Ernest Eugster

Ernest Eugster Ph.D., Thesis Advisor

Shari A. Plantz-Masters

Shari Plantz-Masters

Douglas I. Hart

Douglas I. Hart

Abstract

As the field of Engineering has expanded, researchers and practitioners have shown increasing interest in the role of high quality Requirements Engineering (RE) in the System Development Life Cycle (SDLC) and its impact in determining project success. Traditionally, the literature has been dominated by an effort to establish a wider acceptance of the scenario based approach. New ideas, however, are emerging within the past decade which shows researchers presenting various ways that narrative storytelling might be applied to the scenario based approach. This project contributes to the latest wave of literature that looks at narrative and the scenario based approach to requirements. It examines how screenwriting techniques complementary to the Cooperative Requirements Engineering With Scenarios (CREWS) framework could create advantages when building essential scenarios for requirements elicitation. It shows how screenwriting can be a critical solution technology used in the requirements task of elicitation. These findings verify B. Norden's (2007) previously unproven claim that screenwriting techniques can be used in a Requirements Engineering process. This study, for the first time, compiles the work of the two leading screenwriting authorities R. McKee (1997) and S. Field (2005), showing that there is a coherent screenwriting process. Using the well established CREWS framework, the results show that screenwriting methods are a viable way to generate elicitation scenarios.

Table of Contents

ABSTRACT V

LIST OF FIGURES IX

LIST OF TABLES X

CHAPTER 1 - INTRODUCTION 1

REQUIREMENTS ARE IMPORTANT 1

ADDING VALUE TO THE EXISTING BODY OF KNOWLEDGE 3

PROBLEM STATEMENT 4

PLAN FOR RESEARCHING THE TOPIC 4

SUMMARY 5

CHAPTER 2 - LITERATURE REVIEW 6

PROTOTYPING AND SCENARIOS 6

SCENARIOS AND REQUIREMENTS 7

TOWARD A WIDER ACCEPTANCE OF THE SCENARIO BASED APPROACH 8

EXPLORING NEW AREAS IN THE SCENARIO BASED APPROACH 11

SUMMARY 12

CHAPTER 3 - METHODOLOGY 14

DESCRIBING THE SCREENWRITING PROCESS 14

CREWS: A TOOL TO EVALUATE THE SCREENWRITING PROCESS 14

FORM VIEW 16

CONTENTS VIEW 17

PURPOSE VIEW 20

LIFE CYCLE VIEW 22

SUMMARY 23

CHAPTER 4 – RESULTS 25

DESCRIPTION OF THE SCREENWRITING PROCESS 25

SCREENWRITING PROCESS AND THE FORM VIEW 30

SCREENWRITING PROCESS AND THE CONTENTS VIEW 31

SCREENWRITING PROCESS AND THE PURPOSE VIEW 34

SCREENWRITING PROCESS AND THE LIFE CYCLE VIEW 35

SUMMARY 36

CHAPTER 5 - DISCUSSION 38

FORM VIEW COMPARISON 38

CONTENTS VIEW COMPARISON 39

PURPOSE VIEW COMPARISON 41

LIFE CYCLE VIEW COMPARISON 42

SUMMARY 43

CHAPTER 6 - CONCLUSION 44

MAIN FINDINGS AND CURRENT VIEWS 44

PROBLEMS, ISSUES, AND THE COURSE OF THE RESEARCH 46

FUTURE WORK 46

REFERENCES 47

List of Figures

FIGURE 1 FOUR VIEWS OF SCENARIOS IN THE CREWS FRAMEWORK (ACHOUR ET AL., 1996) 15

FIGURE 2 THE SCREENWRITING PROCESS ACCORDING TO MCKEE AND FIELD............................... 26

List of Tables

TABLE 1 FORM VIEW (ACHOUR ET AL., 1996)........ 16

TABLE 2 CONTENTS VIEW (ACHOUR ET AL., 1996)........ 18

TABLE 3 PURPOSE VIEW (ACHOUR ET AL., 1996)........ 21

TABLE 4 LIFE CYCLE VIEW (ACHOUR ET AL., 1996)........ 22

TABLE 5 FORM VIEW OF THE SCREENWRITING PROCESS........ 30

TABLE 6 CONTENTS VIEW OF THE SCREENWRITING PROCESS........ 31

TABLE 7 PURPOSE VIEW OF THE SCREENWRITING PROCESS........ 34

TABLE 8 LIFE CYCLE VIEW OF THE SCREENWRITING PROCESS........ 35

TABLE 9 FORM VIEW COMPARING INQUIRY ANALYSIS TO SCREENWRITING........ 38

TABLE 10 CONTENTS VIEW COMPARING INQUIRY ANALYSIS TO SCREENWRITING........ 39

TABLE 11 PURPOSE VIEW COMPARING INQUIRY ANALYSIS TO SCREENWRITING........ 41

TABLE 12 LIFE CYCLE VIEW COMPARING INQUIRY ANALYSIS TO SCREENWRITING........ 42

Chapter 1 - Introduction

This project contributes to the latest wave of literature that looks at narrative and the scenario based approach to requirements engineering. Traditionally, the literature was dominated by an effort to establish a wider acceptance of the scenario based approach. New ideas are emerging within the past decade which shows researchers presenting various ways that narrative storytelling might be applied to the scenario based approach. This chapter examines why requirements are important, identifies a problem statement, and introduces the research methodology that will be used.

Requirements Are Important

High quality Requirements Engineering (RE) is important early on in the System Development Life Cycle (SDLC) because it is the standard that determines project success. No systems project should begin without identifying what the system should accomplish, and why the system should satisfy these goals. Easterbrook and Nuseibeh (2000) claimed that the primary measure of success of a system is the degree to which that system meets its intended purpose. That primary measure is determined by defining requirements, and these requirements are found through various elicitation activities.

Royce (1970) recognized decades ago that a lack of requirements analysis resulted in mismanagement, wasted resources, and failure of systems projects to be delivered on time and with success. In 2006, the Standish Group Report showed that 46% of the software projects started that year had cost overruns, time overruns, or did not fully meet the user's needs (as cited in Rubenstein, 2007). The Microelectronics and Computer Technology Corporation (MCC), was a research consortium that studied the problems of designing large software systems by interviewing personnel from several large projects. In their study, the MCC found fluctuating and conflicting requirements caused problems on every project (Curtis, Krasner, & Iscoe, 1988). Sometimes the needs of a single customer changed over time. In other cases, the requirements were defined for the first customer to place an order, even though other customers stated different requirements. On other projects, internal marketing departments added requirements in direct conflict with customer requirements.

According to Linberg (1999), there was one large software project where unrealistic requirements caused the original team leaders to abandon the project. The project was ill defined and fell months behind schedule. This led to extensive overtime so that developers could make code changes as new requirements arose sporadically and without direction. The project was eventually completed with a two year delay and exorbitant cost overruns.

Sumner (1999) described another case where the Boeing airline company resorted to modifying the company's business rules to avoid cost overruns and project failure for their new payroll system. Boeing decided to integrate a standard PeopleSoft package with the legacy payroll system. Boeing found that it was too difficult and time consuming to bring these applications together. The solution was to change the company's business practices to match the limits of the software. Boeing learned that if software needs to be modified then an agreement needs to be made from the start, between IT and management groups, with regards to requirements.

No systems project should begin without identifying what the system should accomplish, and why the system should satisfy these goals. Elicitation encompasses all of the initial activities of RE. Elicitation activities need improved precision, accuracy, and variety of details. Atlee and Cheng (2007) stated that this is the reason requirements elicitation research focuses on technology that improves precision, accuracy and detail. RE is a multi-disciplinary process and the tools and techniques used in RE draw upon a variety of disciplines (Easterbrook & Nuseibeh, 2000).

Adding Value to the Existing Body of Knowledge

Over the last decade, researchers have published breakdowns of different elicitation activities. Davis and Hickey (2003) believed elicitation is about learning, uncovering, extracting, surfacing, and discovering the needs of potential stakeholders. Wiegers (2003) identified the following elicitation activities: user classes, select the product champions, identify the use cases, and identify the system events and responses. Sommerville and Sawyer (2004) listed: identify and consult system stakeholders, collect requirements from multiple viewpoints, prototype poorly understood requirements, and use scenarios to elicit requirements. Finally, Atlee and Cheng (2007) described the requirements elicitation as: identifying stakeholders, refining requirements, eliciting feedback on early representations of the proposed system, and modeling to explore the stakeholders' needs.

These studies show that in the elicitation process, a need to communicate exists with the stakeholders to discover the requirements. An examination of these studies also shows that feedback systems, models, Use Cases, and scenarios are a primary means of communication between the stakeholder and developer. Atlee and Cheng (2007) described Use Cases and scenarios as informal and intuitive exploratory feedback models. Therefore, any reference to the term feedback or modeling also refers to Use Cases and scenarios. Furthermore, Antón, Potts, and Takahashi (1994A) categorized scenarios as merely specific instances of the Use Case. Therefore, any reference to a Use Case implies the more specific instance of a scenario. It can then be inferred from this selection of research studies that the experts perceive scenarios as a critical tool used in elicitation.

This study continues the practice of treating RE as a multi-disciplinary process and it draws from the discipline of screenwriting. Norden (2007) claimed that although a requirements engineer's aim in writing a scenario might not be the same as that of a screenwriter developing a script, they are similar in many ways and screenwriting might improve RE. Although not a rigorous research study, Norden broke down the elements of a screenplay and techniques in screenwriting that can be applied to a requirements process. This study verified B. Norden's (2007) previously unproven claim that screenwriting techniques can be used in an RE process.

Problem Statement

Screenwriting techniques complementary to the Cooperative Requirements Engineering With Scenarios (CREWS) framework creates advantages when building essential scenarios for requirements elicitation.

Plan for Researching the Topic

This is multi-disciplinary study that looks into the screenwriting process, and also examines how requirements elicitation activities use scenario based approaches. It takes in all of the complexities that these separate disciplines exhibit, and then interprets how one benefits from the other. This study interprets the screenwriting process from a Requirements Engineering (RE) perspective. The two questions that need to be examined are:

- Is there a definite screenwriting process?
- How can this process be assessed in terms of a requirements perspective?

To describe the screenwriting process a literary analysis was done on Field's (2005*)* *Screenplay: The Foundations of Screenwriting* and McKee's (1997) *Story: Substance, Structure, Style, and the Principles of Screenwriting*. The results from this research study include a flow chart that describes the screenwriting process. A thorough review of all the literature related to scenarios and RE over the past thirty years, attempts to show that there are tools established by previous researchers to evaluate RE scenario based approaches. The CREWS is a tool designed to compare scenario based approaches used in the requirements process. The screenwriting process is placed upon the CREW framework for analysis.

Summary

This chapter showed that requirements are important. No systems project should begin without identifying what the system should accomplish, and why the system should satisfy these goals. The heart of RE was shown to be elicitation because it is the activity focused on the stakeholder and the definition of their needs. Furthermore, Atlee and Cheng (2007) noted that elicitation is so important to the creation of requirements, that there will seemingly always be a need to improve the precision, accuracy, and variety of details that come out of the elicitation process. The technique of scenario based approaches was introduced as a primary means of communication between the stakeholder and developer. These included: feedback systems, models, Use Cases, and scenarios. Screenwriting techniques, complementary to the CREWS framework, can create advantages when building essential scenarios for requirements elicitation. This chapter continued the established practice of treating RE as a multi-disciplinary process by examining how screenwriting can be used in requirements elicitation. Finally, the screenwriting process through the literary analysis of Field and McKee was introduced. The resulting screenwriting process was then placed upon the CREWS framework to evaluate the RE scenario based approach. Next, we turn our attention to a review of academic and trade literature.

Chapter 2 - Literature Review

This chapter discusses the earliest research into prototyping and scenarios and the recognition of the value these tools have in the field of systems design. Scenarios are particularly valuable in the requirements process where they are used in a number of different activities. More recent research shows a wider acceptance of the scenario based approach and the focus is on categorizing and choosing between the array of techniques and methods of using scenarios. This review also explores new areas of research in the scenario based approach.

Prototyping and Scenarios

Scenarios are an informal and intuitive exploratory model designed to generate early feedback from stakeholders. Scenarios are the stories of what a system is meant to do. Scenarios are a type of modeling that is informal, intuitive, and generates feedback from stakeholders. Scenarios are not tied to any particular method and might include text, graphics, or even be the precursor to a prototype of the system being proposed.

Since the late 1970’s and early 80’s, systems developers have recognized the value of prototyping. Gomaa and Scott (1981) in ”Prototyping as a Tool in the Specification of User Requirements” indicated that a good time to develop a prototype is after an initial version of the requirements specification because the developer has a solid understanding of the problem and has made the first attempt to satisfy the user requirements. However, Gomaa and Scott made no suggestions on how to create this preliminary set of requirements. One can only infer that another process is needed to model these preliminary requirements quickly and inexpensively. Scenario based approaches are informal, intuitive, and inexpensive ways to generate feedback from stakeholders. Scenarios are not tied to any particular method and could be used as the precursor to a prototype of the system being proposed.

Hooper and Hsia (1982) were the first to propose using scenarios to identify requirements and they develop a methodology to identify the requirements by using scenarios that served as a fast initial prototype of the intended system. Chen et al. (1994) revisited this idea in the follow up work "Formal Approach to Scenario Analysis Found" which explored a systematic way to use scenarios in requirements analysis using a formal mathematical base, generating precise scenarios, accommodating change, and keeping users involved in the process. Kyng (1995) was instrumental in framing the relationship between scenarios and prototypes. His research in cooperative design emphasized using low tech tools like scenarios to support design. Scenarios are developed for the end users as a prototype that will contribute to the ongoing design work.

Scenarios and Requirements

The earliest reference to the use of scenarios specifically for the task of elicitation was by Holbrook (1990), who described a methodology that uses early interaction between users and designers to quickly develop a set of initial requirements using scenarios. Holbrook's research provided an overview of how scenarios can be used to refine goals and establish requirements. Antón, Potts, and Takahashi (1994B) "Inquiry-Based Scenario Analysis of Systems Requirements" proposed the Inquiry Cycle model which describes how scenarios are represented as goal-directed plan executions thus providing a bridge from requirements in the planning phase to requirements in the analysis phase. Antón, Potts, and Takahashi (1994A) further explored the Inquiry Cycle model as a formal structure for describing discussions about requirements and requirements activities including: elicitation, documentation, and refinement. These researchers used a case study to demonstrate the Inquiry Cycle model, and to show how scenarios improved requirements analysis. By the mid 1990's, Potts made a direct connection between narrative stories and scenario modeling and proposed a way to apply narrative ideas to user needs. Potts

(1995) in “Using Schematic Scenarios to Understand User Needs” contributed guidelines that define salient scenarios and suggests how they might be used in an interactive system. This paper also outlined a method for writing and using scenarios by analyzing system and user goals.

By the second half of the 1990’s, there were a multitude of methods existed for utilizing scenarios in the entire requirements process. Pohl’s (1997) encyclopedic work proposed a comprehensive framework for characterizing the requirements process: the four worlds of development, subject, system, and usage; and three dimensions which include agreement, representation, and specification. Pohl’s research was furthered by the Cooperative Requirements Engineering with Scenarios (CREWS) report. The most important researchers in the requirements field collaborated on developing a framework to organize all requirements scenario methods into a unified and coherent body of work.

Achour et al. (1996) explored the issues underlying scenario based approaches in requirements and proposes a framework for their classification comprised of: (a) the form view, the contents view, the purpose view, and the lifecycle view; (b) an associated a set of facets which characterize and classify a scenario; (c) facets are measured by a set of relevant attributes. According to Alexander and Maiden (2004) the CREWS project created a framework to classify scenario approaches and inform further research and development work in scenario-based systems.

Toward a Wider Acceptance of the Scenario Based Approach

CREWS research was instrumental in heralding the use of the scenario based approach early on in a system design process. In the Grosz and Rolland (1999) case study of the CREWS L'Ecritoire approach to eliciting requirements showed that scenarios are useful for eliciting requirements, helping in the discovery of exceptional cases, deriving conceptual models, and reasoning about design decisions. With the introduction of the CREWS framework an important connection was made between scenarios and the ubiquitous Use Case design model. Booch, Jacobson and Rumbaugh (1999) emphasized the Use Case design model in the Unified Software Development Process. Jacobson described the Use Case as a story or sequence with a name, an actor, and some text describing a way to use a system (as cited in Alexander, 2004). Achour and Rolland (1998) in "Guiding the Construction of Textual Use Case Specifications", described an approach for transforming partial natural language descriptions of scenarios into well structured and integrated Use Case specifications. Their paper is a useful example of how scenario based design and Use Case design can complement one another.

Up until the late 1990's, there were numerous proposals for how to use scenarios in all phases of the requirements process, not just elicitation. The end of the decade and the beginning of the new millennium was a time when researchers attempted to organize and classify all of the previous research in requirements and scenarios. Many recent studies aimed to organize different ways to use scenarios and put them into a coherent body of knowledge. Sommerville and Sawyer (1997) published *Requirements Engineering: A Good Practice Guide* which gave advice on improving the requirements engineering process, creating requirements which were easier to understand, and creating requirements with fewer errors. Their book was organized into three parts, including: (a) the introduction which discussed the problems of requirements engineering; (b) the guidelines section which made suggestions for improving requirements engineering processes; (c) the final section which contained detailed information to on system modeling, formal methods and viewpoint-oriented approaches. This lengthy volume covered a lot of ground with an emphasis on multiple viewpoint requirements. Easterbrook and Nuseibeh (2000) "Requirements engineering: A roadmap" paper presented an overview of the field of software systems requirements. The strength of this paper was the explanation of core requirements concepts and activities. It summarized older requirements research from the decade of the 1990's.

In a similar work Carroll (2000) in *Making Use: Scenario-based Design of Human-Computer Interactions* made the claim that there is no way forward with design without using some variation of scenario planning. The book attempted to answer questions of what scenario-based design is, how it works, and where it is evolving as a design practice. Even though the thrust of this book is human computer interactions, it had a wealth of valuable information on scenario based design. Also, it covered broad questions about where scenario based design best fits in the management of complexity and ambiguity. Hertzum (2003) reiterated that scenarios have gained acceptance in both research and industry and he advocated more studies involving real-world projects as a way to evaluate current approaches and advance the general understanding of what scenarios contribute to design.

Davis and Hickey (2003) introduced a new unified model of requirements elicitation that showed the critical role of knowledge and technique selection. Their unrealized hope was that this formal model of elicitation would become the standard practice among researchers and practitioners. A different perspective on requirements elicitation modeling emphasized position in the life cycle as the key to technique selection. Alexander and Maiden (2004) predicated their book on the idea that systems complexity can be managed if system design needs are defined early and carefully. They felt the scenario is one of the most powerful techniques for discovering and communicating requirements. Organized around the CREWS framework, this book covered scenario based design across many disciplines and fields. The book was broad in scope and the relevant topics were covered in depth and included: (a) scenarios and systems development; (b) scenarios in requirements discovery; (c) scenarios for innovation; (d) authoring use cases; (e) agile software development; (f) evaluating scenarios; (g) putting scenarios into practice.

Other researchers, namely, Carroll and Go (2004) suggested unifying all the methods of scenario planning into a hierarchy based on the four communities that employ this method of design which include: (a) strategic planning; (b) requirements engineering; (c) human computer interface design; (d) object oriented design. This article surveyed the history and typical scenario usage in different fields, and demonstrated the importance of scenario based approaches in systems development. Atlee & Cheng (2007) reviewed Requirements Engineering (RE) research over the previous decade concentrating on identifying future research directions based on current needs in the software engineering field. This paper provided an overview of the RE field which includes: (a) the research that has been conducted recently; (b) where more effort needs to be focused; (c) the importance of scenario based design.

Exploring New Areas in the Scenario Based Approach

Within the past decade, researchers have presented various ways that narrative storytelling might be applied to the scenario based approach to requirements. Nielson (2002) focused on scenario writing character descriptions that inform the future use of a web site or a system. The researcher's purpose was to look at the scriptwriting process and focus on some of the methods and tools used. Nielson's work looked at scenarios written by experts, letting the inspiration from film scriptwriting suggest ways in which character descriptions could improve scenarios.

Hobbs and Potts (2000) described a computational representation for narrative to aid those who need to plan for multiple contingencies. This unique paper included a thorough break down of terms and concepts from the screenplay that are applicable to scenario based design used in computer modeling. Hobbs (2005) further explored this idea for a computer application framework that allows the use of scenarios for any generalized purpose in "A Scenario-Directed Computational Framework to Aid Decision-Making and Systems Development".

Strom's (2007) paper described how stories with emotions and conflicts were used to help define requirements where the most useful stories were found to have the following: (a) plots driven by realistic conflicts and emotions; (b) a focus on the actions more than the emotions of the participants; (c) internal consistency; (d) realistic dialogue; (e) detailed descriptions of situations and the work environment. This study showed that human centered stories are easy to read, they involve the reader, and they make it possible for the reader to experience what the user goes through to reach their goal. However, the study also showed that there is a need for more concise and structured technical descriptions during the development process, and it is not feasible to use stories as the only working papers during the actual software development.

Norden (2007) extrapolated advice for requirements engineers from the process of screenwriting using the following: the synopsis, the scene by scene, action, story beats, sequences, character goals, and story arc. Rinzler's (2009) book explains how to write a software requirements document using the following characteristics of storytelling: (a) narrative that is engaging and easy to follow; (b) problems clearly stated at the beginning; (c) story points follows from one to the next in a logical sequence of outcomes; (d) all points lead to the solution. Rinzler and others established a precedent for research that examined requirements in relation to narrative stories, screenwriting, and scenario based approaches.

Summary

This review discussed the earliest research into prototyping and scenarios and the recognition of the value these tools have in the field of systems design. Since the late 1970's and early 80's, systems developers have recognized the value of prototyping. Scenario based approaches were informal, intuitive, and inexpensive ways to generate feedback from stakeholders. Scenarios were not tied to any particular method and could be used as the precursor to a prototype of the system being proposed. Scenarios were particularly valuable in the requirements process where they were used in a number of different activities.

By the second half of the 1990's, there were a multitude of methods for utilizing scenarios over the entire requirements process. In that decade, the most important researchers in the requirements field collaborated on the CREWS proposal, which suggested a framework to organize all requirements scenario methods into a unified and coherent body of work. CREWS research was instrumental in heralding the use of the scenario based approach early on in a system design process. Recent research showed a wider acceptance of the scenario based approach and a focus on categorizing and choosing between the array of techniques and methods of using scenarios.

At the beginning of the new millennium, researchers attempted to make sense of all the previous research in requirements and scenarios. New ideas emerged within the past decade which showed researchers presenting various ways that narrative storytelling might be applied to the scenario based approach. However, none of the most recent writings described a rigorous research study that demonstrated how the screenwriting process can specifically be of use in a requirements elicitation process.

Chapter 3 - Methodology

This chapter develops the methodology to answer the questions that need to be examined. This is multi-disciplinary study examines the screenwriting process, and how requirements elicitation activities use scenario based approaches. It takes in all of the complexities that these separate disciplines exhibit, and then interprets how one benefits from the other. This study interprets the screenwriting process from a Requirements Engineering (RE) perspective. The two questions that need to be examined are:

- Is there a definite screenwriting process?
- How can this process be assessed in terms of a requirements perspective?

Describing the Screenwriting Process

This study investigates viability of using screenwriting process for creating requirements elicitation scenarios. A literary analysis was done on Field's (2005) *Screenplay: The Foundations of Screenwriting* and McKee's (1997) *Story: Substance, Structure, Style, and the Principles of Screenwriting*. The results from this research study included a flow chart that describes the screenwriting process. The screenwriting process came about through the aforementioned literary analysis.

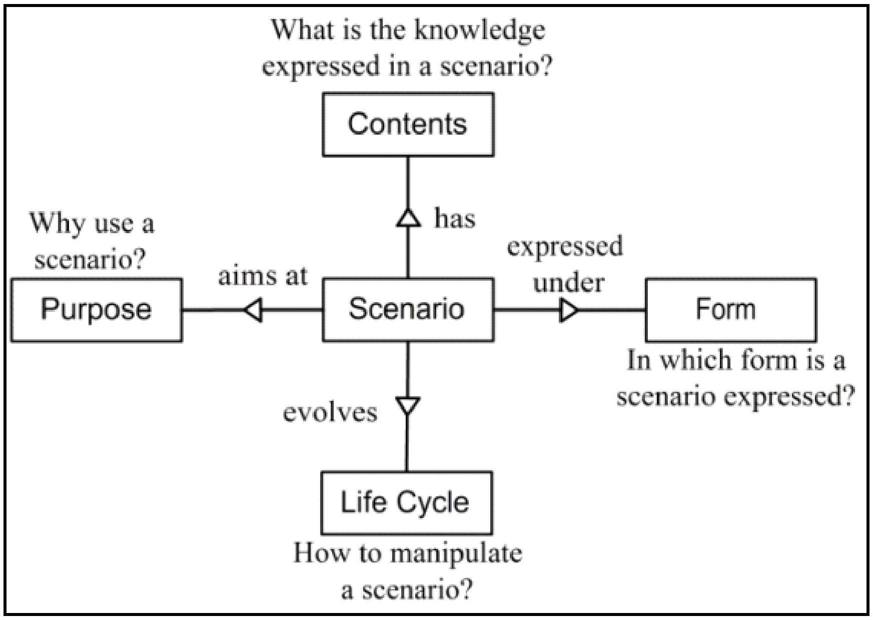
Flow charts have been used to increase the effectiveness of communication among stakeholders and developers for decades. Flow charts provide an orderly representation of data and a sequence of operations. A flow chart is simply a diagram of how different stages in an operation, in this case screenwriting, are interconnected. This research project complied with the requirements of the International Organization for Standardization (ISO) Recommendation on Flowchart Symbols for Information Processing ("Flow Charting", 1969).

CREWS: A Tool to Evaluate the Screenwriting Process

A thorough review of all the literature related to scenarios and Requirements Engineering over the past thirty years, showed that there are tools established by previous researchers to evaluate RE scenario based approaches. The Cooperative Requirements Engineering With Scenarios (CREWS) is a tool designed to compare scenario based approaches used in the requirements process. According to Achour et al. (1996), the reasons for developing the framework were: (a) to help understand and clarify scenario based approaches; (b) to gain a perspective on the industrial practice of scenarios; and (c) to associate common RE situations to types of scenarios. CREWS provided a framework for classifying and reviewing the screenwriting process as scenario based approach to RE elicitation.

The CREWS framework considers different scenario approaches through four different *views*, each capturing *facets* of the scenarios. As seen in Figure 1, the framework consists of four views: *Form*, *Contents*, *Purpose*, and *Lifecycle*. Each view has different facets with each facet defined by several *attributes*. A scenario based approach is positioned in the framework by assigning a value to each attribute of each facet. Attribute values are defined within a set. A set may be a Boolean type or a set of values that are established by the inventors of CREWS. Unless otherwise noted, the Boolean value “True” indicates whether an approach supports the facet’s attribute, and the “False” value indicates that the approach does not.

Figure 1 Four Views of Scenarios in the CREWS Framework (Achour et al., 1996)


What is the knowledge expressed in a scenario?
Contents
has
Why use a scenario?
aims at
Purpose
Scenario
expressed under
Form
In which form is a scenario expressed?
evolves
Life Cycle
How to manipulate a scenario?

Form View

The form view documents how scenarios are expressed. The form view contains two facets: *description* and *presentation*.

Table 1 Form View (Achour et al., 1996)

Facet	Attribute	Values
Description	Medium	{Text, Image, Graphics, Prototype}
	Notation	{Any, Informal, Semi-Formal, Formal}
Presentation	Animation	{True, False}
	Interactivity	{ None, Hypertext-like, Advanced}

Description. This facet covers how a scenario is created and its appearance. The description facet has two attributes:

1. *Medium* - described using the set of values: (a) text, (b) graphics, (c) image, (d) video, and (e) software prototype. In general, scenarios are associated with the

narration of stories or narrative text. Formatted text includes tables or scripts. Graphics, images, and video are other types of media that can also be used. Scenarios may also be presented as systems using prototypes, mock-ups, or simulators.

2. *Notation* - (a) Formal scenarios can be defined with predetermined symbols; (b) Semi-formal may use natural language in a structured format such as tables; (c) Informal notation is defined as scenarios that make the use of natural language.

Presentation. This facet covers how a user might experience and control a scenario. The description facet has two attributes:

1. *Animation* - has the Boolean value which equates to static or animated. The animation attribute specifies whether or not there are capabilities for visualizing the system on screen.
2. *Interactivity* - relates to the control offered to the user to progress the scenario through time. If interactivity is provided, the user can act at various stages of the scenario presentation: (a) None refers to the interaction with the user; (b) Hypertext-like links can be inserted in scenarios; (c) Advanced interaction gives the user the choice to modify the flow of an animation by triggering actions and events.

Contents View

The Contents view documents the kind of knowledge that is expressed in a scenario. Acour et al. (1996) made a distinction between scenarios which broadly describe the world at the level of people and scenarios which describe the detailed behaviors and dependencies of a system. The Contents view has four facets: *Abstraction*, *Context*, *Argumentation*, and *Coverage*.

Table 2 Contents View (Achour et al., 1996)

Facet	Attribute	Values
Abstraction	Instance	{True, False}
	Type	{True, False}
	Mixed	{True, False}
Context	System Internal	{True, False}
	System Interaction	{True, False}
	Org. Context	{True, False}
	Org. Environment	{True, False}
Argumentation	Position	{True, False}
	Arguments	{True, False}
	Issues	{True, False}
	Decision	{True, False}
Coverage	Functional	{Structure, Function, Behavior}
	Intentional	{Goal, Goal decomposition, Responsibilities, Opportunity }
	Non-functional	{Performance, Time/Cost Constraints, User Support, Flexibility, Error Handling}

Abstraction. Does the scenario describe actors and events at the type level, instance levels, or both? The three attributes of the facet: *Instance*, *Type*, and *Mixed* allow a measure of the level of abstraction or concreteness depending upon the contents of a scenario based approach. It is difficult to determine the level of abstraction for informal scenarios which contain complex situations (Achour et al., 1996). The three attributes may be assigned the value "True" if the approach accepts scenarios containing both instance and type information:

1. Instance - concentrates on details of individual actors, events, stories, and episodes with little or no abstraction.
2. Type - more abstract scenarios. Type scenarios describe facts in broad categories.
3. Mixed - contain different levels of abstraction. Information is described at both the Instance and Type levels.

Context. The context facet aims at classifying scenario approaches according to the amount of existing condition information they capture. There are four attributes attached to the context facet:

1. *System Internal* - internal behavior of the system.
2. *System Interaction* - interactions with its environment.
3. *Organizational Context* - broad picture of how the work gets done including knowledge on the stakeholders: their motivations, goals, social relationships, membership in groups, and responsibilities.
4. *Organizational Environment* - the widest perspective where an organization is itself influenced by external factors such as: history, legislation, and economics.

Argumentation. Knowledge pertaining to why a system has certain features can be captured by a scenario. What are the reasons why certain actors perform certain actions? Does the scenario support different types of justification? The Argumentation facet is broken down into the attributes:

1. *Position* - descriptions of alternative solutions to a problem.
2. *Arguments* - for objecting or supporting a given position.

3. *Issues* - descriptions of problems or conflicts.
4. *Decision* - choices of a particular position.

Coverage. What sets of things does the scenario cover? The Coverage facet aims at classifying scenario based approaches according to the kind of information they capture. Typical contents include: structure of a company, groups, departments, agents, and stakeholder. Other types of content are the characteristics of people, including: their views, aspirations, wishes, aims, and objectives. Coverage is described by the following attributes:

1. *Functional* - a complete description of an object can be made using a three level representation: (a) *Structure* - shape, components, configuration, material, and surface finish; (b) *Behavior* - how many states, state-state changes, time taken, flow rates, etc.; (c) *Function* - what useful behaviors will be available in what environments under what situations.
2. *Intentional* - understanding of an organization's objectives, intentions and goals. Based on the analysis of the goal driven approaches, the refined intentional attribute uses the following set: (a) *Goal*; (b) *Problem*; (c) *Responsibility*; (d) *Opportunity*; (e) *Cause*; (f) *Goal dependency*.
3. *Non-functional* - guidelines about what kind of non-functional requirements that should be expressed, and how to express them. Non-functional coverage is further refined in the framework with: performance, time constraints, cost constraints, user support, documentation, examples, backup / recovery, maintainability, flexibility, portability, security / safety, design constraints, and error situations.

Purpose View

The Purpose view documents the role that a scenario plays in the requirements-engineering process. There are core purposes for using scenarios that the Purpose view tries to identify. Along the Purpose view, scenarios are classified according to the role they aim to play in the requirements engineering process. Purpose view is comprised of three attributes: *Descriptive*, *Exploratory* and *Explanatory*. The same scenario can be used for several purposes. Descriptive and Exploratory scenarios must be investigated by means of an inquiry process, while Explanatory scenarios are given more spontaneously.

Table 3 Purpose View (Achour et al., 1996)

Facet	Attribute	Values
Role	Descriptive	{True, False}
	Exploratory	{True, False}
	Explanatory	{True, False}

Role. The Purpose view has only one facet. The facet Role has three attributes defining whether or not the scenario fulfills a:

1. Descriptive role - the functionality of a system. This is walking through a process to understand its operations, involved actors, and triggering events. Requirements can be derived from descriptions of one or more transactions involving the system and its environment.
2. Exploratory – scenarios that are useful when several different possible solutions for satisfying a system requirement have to be explored and evaluated. These scenarios make the link between solutions and requirements.

3. Explanatory - these scenarios provide detailed illustrations of the situation and the rationale. These scenarios define the cause of drawbacks, inefficiencies, and a lack of system performance.

Life Cycle View

The Life Cycle view documents scenarios as artifacts that are created, refined, or deleted over time in a requirements process. How scenarios are captured, evolve, and are improved is the concern of the life cycle view. It has two facets: *Lifespan* and *Operation.*

Table 4 Life Cycle View (Achour et al., 1996)

Facet	Attribute	Values
Lifespan		{Persistent, Transient}
Operation	Capture	{Reuse, Scratch}
	Integration	{True, False}
	Refinement	{True, False}
	Expansion	{True, False}
	Deletion	{True, False}

Lifespan. The Lifespan facet has one attribute, with the values Transient and Persistent based on whether the scenario is transient or persistent in the systems life cycle. Transient scenarios are meant to be a support for some requirements activity and are thrown out after being used. Persistent scenarios exist as long as the documentation of the project they belong to exists. There are two reasons for scenarios to be persistent: (a) scenarios are considered part of the requirements specification; (b) the project documentation keeps track of the scenarios used.

Operation. As any dynamic artifact, scenarios are created, transformed and deleted through the execution of operations. The operation facet aims at classifying scenarios according to the kinds of operations carried out on them. Thus, this facet is concerned with how scenarios are captured, evolve and are eventually transformed during the requirements process. The operation facet has five facets:

1. *Capture* - operations deal with the generation of scenarios. Almost all approaches create scenarios from scratch.
2. *Integration* - scenarios can be thought of as stories which are fragmented (Achour et al., 1996). If produced as fragmentary pieces of details, scenarios can be integrated.
3. *Refinement* - transformation of scenarios to make them easy to understand or more reusable. A re-structure of scenarios without increasing their contents.
4. *Expansion* - adds new knowledge in a scenario description.
5. *Deletion* - terminates the scenario lifespan.

Summary

This chapter looked into the screenwriting process and examined how this process can be applied to requirements elicitation. Using the work of McKee and Field it is possible to make sense of the screenwriting process. Incorporating their ideas about the screenwriting process into a flow chart provided an orderly representation of data and a sequence of operations. The flow chart was simply a diagram of how different stages in screenwriting were interconnected.

This study interpreted the screenwriting process from a Requirements Engineering (RE) perspective using the CREWS framework. CREWS is a tool established by previous researchers to evaluate RE scenario based approaches. CREWS compared scenario based approaches used in the requirements process. The framework considered different scenario approaches along four different views, each capturing facets of the scenarios. The framework consisted of four views: Form, Contents, Purpose, and Lifecycle. To each view belonged different facets. Each facet was defined with several attributes. A scenario based approach was positioned in the framework by assigning a value to each attribute of each facet. CREWS informed this research project by providing a framework for classifying and reviewing the screenwriting process as scenario based approach to RE elicitation.

Chapter 4 – Results

As seen in Chapter 3, this study interprets the screenwriting process from a Requirements Engineering (RE) perspective using the CREWS framework. Before using the framework the screenwriting process needs to be outlined. These results contain a flow chart which describes the screenwriting process including a description of the flow chart elements and the justification for their inclusion. Once the screenwriting process is outlined, the focus of the project then turns to analyzing the viability of the screenwriting process for the purpose of requirements elicitation. The CREWS framework evaluates scenario based approaches in Requirements Engineering (RE). This chapter aims to position the screenwriting process upon the CREWS framework and to this end the chapter provides a brief, line by line, rational for the attribute values selected.

Description of the Screenwriting Process

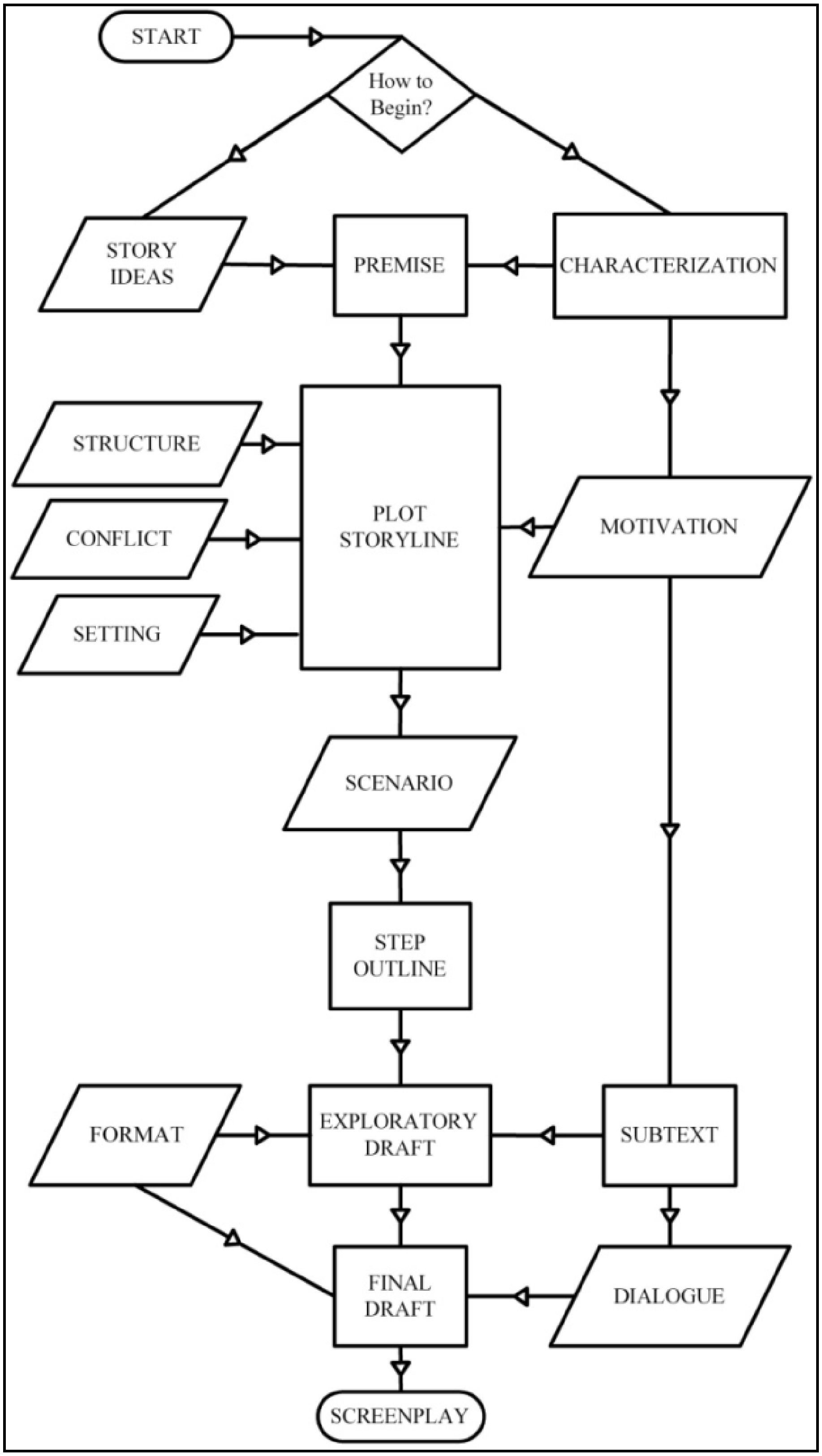
Figure 2 shows a flow chart describing the screenwriting process. This outline is not a definitive screenwriting process because any attempt to define one, and only one, standard creative process is futile. This process, however, is based on a combination of ideas prescribed by Robert McKee and Syd Field.

Field (2005) in *Screenplay: The Foundations of Screenwriting* defined a screenplay as a story told in pictures. He noted that a screenplay has a subject, and is about a main character, in a place, and performing some kind of action. Screenplays have common conceptual components and these elements are expressed dramatically within a structure that has a definite beginning, middle, and end; corresponding to the set-up, the confrontation, and the resolution. Field also summarized the nature of drama: all drama is conflict; with conflict there is action; with action there is character; and with character there is story (Field, 2005).

McKee's (1997) *Story: Substance, Structure, Style, and the Principles of Screenwriting*, considered a standard text in the field of screenwriting, examined films to identify the components of their stories. He defined basic components of film story: beat, scene, sequence, act, and climax; multiple act dramatic structures; importance of theme, setting, and atmosphere; and the importance of character. McKee dissected film scenes revealing why they work, and the fundamentals of composition. He included useful chapters on the principles of story design, scene design, and scene analysis.

This flow chart attempts to organize the ideas of these preeminent experts in screenwriting, and attempts to draw on the salient, common points of both McKee and Field. The writers come to agreement on the same concepts, but at times use a different vocabulary. Figure 2 also shows an attempt to define a standard vocabulary and to describe the components that go into the screenwriting that may be relevant to requirements elicitation scenarios.

Figure 2 The Screenwriting Process According to McKee and Field


START
How to Begin?
STORY IDEAS
PREMISE
CHARACTERIZATION
STRUCTURE
CONFLICT
SETTING
PLOT STORYLINE
MOTIVATION
SCENARIO
STEP OUTLINE
FORMAT
EXPLORATORY DRAFT
SUBTEXT
FINAL DRAFT
DIALOGUE
SCREENPLAY

Start. At the start of the screenwriting process a decision needs to be made: how to begin? The screenwriter must start with *Story Ideas* or through the *Characterization* process. This leads to the *Premise* also known as the central concept or central idea.

Premise. The Premise is a process that takes input from Story Ideas and Characterizaton. Story Ideas are data describing subject matter or theme while Characterization is a process that creates main and supporting characters. McKee (1997) described the Premise as the controlling idea and it may be expressed in a single sentence describing how and why life undergoes change from one condition to another. Field (2005) called the Premise the subject and it can be described in a few sentences, in terms of actions and character.

Characterization. The Characterization process involves generating biography and background stories for new characters. The single most important piece of data to come out of the character process is *Motivation*. It is through Characterization that a character's needs or goals are developed. The resulting Motivation data describe a character's need and goals. Motivation data helps in the *Subtext* process in the later stages of screenwriting, which ultimately result in *Dialogue* data. McKee (1997) described Characterization as the sum of all observable human qualities, including: age, IQ, sex, education, occupation, personality and values. Field (2005) traced the character's life until the story begins, examining career, relationships, dreams, hopes, and aspirations.

Plot Storyline. The *Plot Storyline* process is informed by *Structure*: screenplay data describing acts, set-ups, resolutions, turning points, and inciting incidents. These elements are expressed dramatically within a structure that has acts: a definite beginning, middle, and end. These acts correspond to the set-up, the confrontation, and the resolution. Structure is a selection of events from a character's life stories, composed into a sequence to arouse emotions and to express a view (McKee, 1997). The *Plot Storyline* process adds *Conflict* data including obstacles and tension. These are created in opposition to motivation data. This process also adds *Setting* data which pertains to time and place.

Scenario data result from the plot storyline process. Scenarios are story events that begin as a one or two line description, and Field (2005) recommended establishing purpose then determining the content of the scenario. An event is caused by or affects characters: it takes place in a setting, generating image, action, and dialogue; it draws energy from conflict producing emotion in characters and the audience, so that events must be composed and not random (McKee, 1997). Field (2005) recommended writing the idea of each scene or sequence on a single card with a few brief words of description.

Step Outline. The *Step Outline* is the process of creating an organized sequence of Scenarios. On the back of each card, the writer indicates what step in the design of the story he sees this scene fulfilling (McKee, 1997). The Step Outline includes all the scenes that have to be included. Field (2005) recommended establishing the dramatic need of the main character, then applying the need to each scene, and finally creating the obstacles. The Step Outline of all the scenes is a summary of the action and lays out the story line. McKee and Field both recommended creating an outline or treatment out of the scenario cards.

Exploratory Draft. The exploratory draft is the process of expanding each scene from its one or two sentences into a paragraph or more of present tense and moment by moment description (McKee, 1997). *Format* data is added which informs the screenwriter as to how to organize a screenplay. This data includes: scenes, actions, sound effects, and camera shots. This research study makes a distinction here: Format scene data is not a Scenario. Format scene data is a necessity for the screenplay. In terms of Format scene data the screenplay is divided into scene headings. In the final screenplay, any change in location or lighting will require a new scene heading (Field, 2005). The Scenario, on the other hand, is conceptual. The Scenario is a continuous series of actions with a purpose, but the Scenario is not a component of the screenplay Format data.

Subtext. The process of creating dialogue data is described as *Subtext*. Subtext is a process that primarily informs the E*xploratory Draft*. Forty to sixty scenes of a typical screenplay treated to a moment by moment description of all action under laid with a full subtext of the conscious and unconscious thoughts and feelings of all characters will produce the Exploratory Draft (McKee, 1997). In the Exploratory Draft, the writer indicates what characters talk about but does not write dialogue. Instead Subtext is created, which is defined as the true thoughts and feelings underneath what is said and done (McKee, 1997).

Final Draft. This process is the final rendering of Dialogue and Format data. Dialogue is data resulting from the Subtext process. Dialogue can be traced back to Motivation data, and ultimately the character creation process. The Dialogue data is a function of character: moving the story forward; revealing information about the character; communicating necessary facts; and revealing the conflicts of the story (Field, 2005).

Screenplay. The screenplay is a selection of events from the characters' stories that is composed into a sequence to arouse specific to express a specific ideas and feelings (McKee, 1997). A screenplay may be defined as a linear arrangement of related incidents, episodes, or events that lead to a dramatic resolution, and how these structural components are utilized determines the form of the screenplay (Field, 2005).

Screenwriting Process and the Form View

Table 5 Form View of the Screenwriting Process

Facet	Attribute	Value
Description	Medium Notation	Text Semi-formal
Presentation	Animation Interactivity	False None

Description.

1. Medium - scenarios are described with narrative text.
2. Notation - narrative text is defined as the value “Informal”, however the screenwriting process does result in several Semi-formal artifacts: Scenarios, the Step Outline, Exploratory Draft and Final Draft.

Presentation.

1. Animation - scenarios are statically used.
2. Interactivity - scenarios are used without any possible planned interaction with the users.

Screenwriting Process and the Contents View

Table 6 Contents View of the Screenwriting Process

Facet	Attribute	Value
Abstraction	Instance Type Mixed	True True True
Context	System Internal	True

	System Interaction	True
	Org.Context	True
	Org.Environment	True
Argumentation	Position	True
	Arguments	False
	Issues	True
	Decision	True
Coverage	Functional	Structure, Function, Behavior
	Intentional	Goal, Problem, Responsibility, Cause
	Non-functional	{ }

Abstraction.

1. Instance - screenwriting deals with characters that react to events and obstacles. Therefore, the screenwriting process defines the scenario contents at the Instance level. Instance scenarios, or concrete scenarios, refer to specific agent names or events with concrete argument values.
2. Type - scenarios can be categorized into scenes, sequences, and acts.
3. Mixed - the Type attribute is True because there can be a mix of Instance and Type contents.

Context.

1. System Internal - in the screenwriting process the system can be considered as a character, internal actions can be represented.
2. System Interaction - system interactions with its environment can be described.

3. Organizational Context - screenwriting process can include knowledge concerning motivations, goals, social relationships, membership in groups, and responsibilities.
4. Organization environment - the screenwriting process can show how an organization is itself influenced by external factors such as: history, legislation, and economics.

Argumentation.

1. Position - there could be descriptions of alternative solutions to a problem organized in the Step Outline and Exploratory Draft processes.
2. Arguments - there is no means to record objecting or supporting a given position, even though alternative solutions can be presented.
3. Issues - the Plot Story process and Scenario data must contain descriptions of problems or conflicts within a given situation.
4. Decision - choices of a particular position are seen in the final Exploratory Draft.

Coverage.

1. Functional - a complete description of the final system using the screenwriting process including: (a) Structure - components and configuration can be described using the Exploratory Draft; (b) Behavior - describing changes in time from scene to scene in Step Outline or Exploratory Draft processes; and (c) Function - description of useful behavior and environments are a part of all the Scenario data.
2. Intentional - if the system can be described using the Characterization process, then there has to be some recorded understanding of the system's intentions and

goals. These might include statements about: Goal, Problem, Responsibility and Cause.

3. Non-Functional - aspects such as: performance, time constraints, cost constraints, support, and design constraints are not covered in the screenwriting process and would most likely some other production management process.

Screenwriting Process and the Purpose View

Table 7 Purpose View of the Screenwriting Process

Facet	Attribute	Value
Role	Descriptive	True
	Exploratory	True
	Explanatory	False

Role.

1. Descriptive - screenwriting scenarios rely exclusively upon descriptive scenarios, they are end-end sequences of activities performed by several roles to achieve some goal. System functionality could also be described between the system and its actors.
2. Exploratory - it is possible to present several different possible solutions for satisfying a given system requirement, but there is no means to evaluate and support a decision.
3. Explanatory - screenwriting process discourages explanation in favor of simply presenting actions and dialogue. Thus, it is possible to present several possible solutions, but there is no detailed illustration of the rationale behind them.

Screenwriting Process and the Life Cycle View

Table 8 Life Cycle View of the Screenwriting Process

Facet	Attribute	Value
Lifespan		Persistent
Operation	Capture	Scratch
	Integration	True
	Refinement	True
	Expansion	True
	Deletion	True

Lifespan. Screenwriting scenarios are stored in an artifact document and therefore, are persistent.

Operation.

1. Capture - scenarios are constructed from scratch.
2. Integration - screenwriting scenarios are fragmented stories. They are initially produced as separate pieces of details, and can be integrated.
3. Refinement - transformation of scenarios to make them easy to understand or more reusable. A re-structure of scenarios without increasing their contents. This is done in the Step Outline process.
4. Expansion - adding new knowledge in a scenario description is possible only at the Plot Storyline process where the Scenario data is generated. McKee and Field endorse changes during the Final and Exploratory Draft processes, but recommend going back to the Step Outline or Plot Storyline processes to due so.

5. Deletion - scenarios are used to support the discovery of requirements, and can be thrown immediately after use.

Summary

Before using the CREWS framework, the screenwriting process was outlined. The screenwriting process ultimately resulted in a formatted screenplay with scenes, but this final artifact was not the real value of the screenwriting process. Format data, Dialogue data and a Final Draft were an extraneous part of the screenwriting process with regards to requirements elicitation. It was the opinion of this researcher that removing the screenplay Format data, Dialogue data, and Final Draft yields, a scenario based approach that could be effective in a requirements elicitation process.

This chapter has revealed that the real value of the screenplay approach included all of the data and processes up to, and including, the Exploratory Draft. After a scenario is created using the Plot Storyline process, a Step Outline and Exploratory Draft process would be performed. This chapter revealed that this Exploratory Draft was the most important artifact because it took into account all the data that goes into generating Scenario data, as well as, the Step Outline process, and Subtext data. These processes and data were the most relevant parts of the screenwriting process respective to requirements elicitation.

The focus then turned to analyzing the viability of the screenwriting process for the purpose of requirements elicitation. The CREWS framework evaluated scenario based approaches in RE. When positioning the screenwriting process upon the CREWS framework, this study considered only the data and processes up to and including the Exploratory Draft. The Form view documented how scenarios in the screenwriting process used narrative text, arranged in a semi-formal manner, without animation or interaction. The Contents view also showed that the kind of knowledge expressed in screenwriting scenarios includes Abstraction and Context information. Argumentation facet was limited in the screenwriting process, and so was the Coverage facet. The Purpose view showed that the screenwriting process plays both a Descriptive and Exploratory role. Finally, the Life Cycle view showed that the screenwriting process created, refined, and deleted artifacts over time in a requirements process.

Chapter 5 - Discussion

The CREWS framework can be used to highlight the advantages the screenwriting process has in the requirements elicitation process. By comparing the CREWS results for the screenwriting process with an established CREWS requirements elicitation approach, these highlighted advantages can be better discerned. This discussion employs the CREWS framework to compare the Inquiry Analysis scenario based approach to the screenwriting process. This comparison serves to highlight the merits and weaknesses of the screenwriting process in requirements elicitation. The Inquiry Analysis Approach was selected because it provided a formal structure to support requirements identification following analysis activities from requirements elicitation and documentation through refinement (Antón, Potts, and Takahashi, 1994A). It was also well documented in Achour et al. (1996).

Form View Comparison

Table 9 Form View Comparing Inquiry Analysis to Screenwriting

Facet	Attribute	Potts Inquiry Analysis (Achour et al., 1996)	Screenwriting Process
Description	Medium	Text	Text
	Notation	Semi formal	Semi-formal
Presentation	Animation	False	False
	Interactivity	Hypertext-like	None

In the Inquiry Cycle approach, scenarios are in textual form and placed within tables. One advantage of Inquiry Analysis is the support by a hypertext tool. The screenwriting process has no equivalent hypertext tool. In a screenwriting process, scenarios are described with narrative texts. These scenarios are static, without any possible planned interaction with the users.

Contents View Comparison

Table 10 Contents View Comparing Inquiry Analysis to Screenwriting

Facet	Attribute	Potts Inquiry Analysis (Achour et al., 1996)	Screenwriting Process
Abstraction	Instance	True	True
	Type	True	True
	Mixed	True	True
Context	System Internal	True	True
	System Interaction	True	True
	Org.Context	True	True
	Org.Environment	False	True
Argumentation	Position	False	True
	Arguments	False	False
	Issues	False	True
	Decision	False	True
Coverage	Functional	Structure, Function, Behavior	Structure, Function, Behavior
	Intentional	{ }	Goal, Problem, Responsibility, Cause
	Non-functional	{ }	{ }

Inquiry Analysis supports instance scenarios; however, most situations are expressed at the type level. The screenwriting process, on the other hand, creates most scenarios at the Instance level. Scenarios in the screenwriting process can then be categorized into scenes, sequences, and acts. This allows screenwriting to organize all the instances into Types in the Step Outline process and Exploratory Draft. Both Inquiry Analysis and the screenwriting process can mix Instance and Type contents.

In the screenwriting process and Inquiry Analysis, the system can be considered as a character, internal actions can be represented, and the detail of internal actions can be described through more detailed scenarios. Both scenario based approaches can include knowledge concerning motivations, goals, social relationships, membership in groups, and responsibilities. However, only the screenwriting process describes system interactions with its environment. The screenwriting process can show how an organization is influenced by broad scale external factors such as: history, legislation, and economics. This may be captured in the Characterization or Subtext process, and through Motivation data, Conflict data, or even Setting data.

The CREWS framework indicates that the screenwriting process has a clear advantage in the Argumentation facet. The Inquiry Analysis raises questions which are discussed in an activity separate from requirements elicitation. Inquiry Analysis does not capture the Argumentation facet in any way. The screenwriting process is tailored to express descriptions of alternative solutions to a problem through the Step Outline and Exploratory Draft processes. In addition the Plot Story process and Scenario data must contain descriptions of problems or conflicts within a given situation. Also, choices of a particular position are seen in the final Exploratory Draft. While the screenwriting process can be annotated with rationale or assumptions, i.e. through Subtext, there is no means to record objections or support for a given position.

Finally, Inquiry Analysis scenarios only cover functional requirements of the system, and Inquiry Analysis does not capture goals or intentions. While the screenwriting process cannot capture the Non-functional attribute, the process does capture the Intentional attributes through the Exploratory Draft such as: Subtext process, Motivation, and Conflict data. Intentional Attribute values such as goals, problems, and responsibility and causation would be described in an Exploratory Draft.

Purpose View Comparison

Table 11 Purpose View Comparing Inquiry Analysis to Screenwriting

		Potts Inquiry Analysis	Screenwriting
Facet	Attribute	(Achour et al., 1996)	Process
Role	Descriptive	True	True
	Exploratory	False	True
	Explanatory	False	False

Inquiry Analysis and the screenwriting process rely upon descriptive scenarios. Inquiry Analysis scenarios are end to end sequences of activities performed by several roles to achieve some goal (Achour et al., 1996). They primarily serve the purpose of capturing requirements. The CREWS framework indicates that there is an advantage in the screenwriting process in terms of exploring solutions to a system requirement. Only in the screenwriting process is it possible to present several different solutions for satisfying a given system requirement. There is, however, no means to evaluate and support an argued choice in terms of the Role view.

Life Cycle View Comparison

Table 12 Life Cycle View Comparing Inquiry Analysis to Screenwriting

Facet	Attribute	Potts Inquiry Analysis (Achour et al., 1996)	Screenwriting Process
Lifespan		Persistent	Persistent
Operation	Capture	Scratch	Scratch
	Integration	True	True
	Refinement	False	True
	Expansion	False	True
	Deletion	True	True

The Inquiry Cycle approach and screenwriting process keeps track of the scenarios used. Both approaches store scenarios in the artifact document and, therefore, are persistent. In both approaches, scenarios are constructed from scratch. Also, scenarios from Inquiry Analysis and the screenwriting processes can be integrated. Scenarios are stories which are fragmented in nature, and if initially produced as fragmentary pieces of details, those scenarios can be integrated (Achour et al., 1996). Screenwriting Scenario data can also be gathered into a Step Outline and organized into a progression. The Inquiry Analysis approach does not refer to any refinement operation. The CREWS framework indicates that there are advantages to the screenwriting process in terms of Refinement, Expansion and Analysis: (a) transformation of scenarios to make them easy to understand is done in the Exploratory Draft and Step Outline processes; (b) adding new knowledge in a scenario description is possible at the Plot Storyline process; and (c) McKee and Field endorse changes during the Final and Exploratory Draft processes, but recommend going back to the Step Outline or Plot Storyline processes to due so. Finally, in both Inquiry Analysis and the screenwriting process scenarios are used to support the discovery of requirements, and can be discarded immediately after use.

Summary

Inquiry Analysis was used in this discussion to better discern the advantages gained by using the screenwriting process as an alternative scenario approach. In the Form view both methods employed narrative texts, but only Inquiry Analysis provided interaction with the users. In the Contents view only the screenwriting process described system interactions with its environment. The Contents view also indicated that the screenwriting process has a clear advantage in the Argumentation facet. The screenwriting process covered functional requirements of the system, and captured goals or intentions, where Inquiry Analysis does not. The Purpose view indicated that there is an advantage in the screenwriting process in terms of exploring solutions to a system requirement. Finally, the Life Cycle view showed that there are advantages to the screenwriting process in terms of Refinement, Expansion and Analysis.

Chapter 6 - Conclusion

This researcher believes this project contributes to the latest wave of literature that looked at narrative and the scenario based approach to Requirements Engineering (RE). Traditionally, the literature was dominated by an effort to establish a wider acceptance of the scenario based approach. New ideas emerged within the past decade which showed researchers presenting various ways that narrative storytelling might be applied to the scenario based approach. However, none of the most recent writings described a rigorous research study that demonstrated how the screenwriting process can specifically be of use in a requirements elicitation process. This study aimed to fill the gap by showing that screenwriting techniques complementary to Cooperative Requirements Engineering With Scenarios (CREWS) framework created advantages when building essential scenarios for software requirements elicitation.

Main Findings and Current Views

This study clarified a common definition of requirements elicitation based on the work of several information systems research studies. The heart of RE was perceived as elicitation because it was the activity focused on the stakeholder and the definition of their needs. The scenario based approach was a critical solution technology used in requirements elicitation. Scenario based approaches to requirements elicitation were essential because these approaches were informal, intuitive, and generated early feedback from stakeholders.

This study, for the first time, compiled the work of the two leading screenwriting authorities McKee and Field, and showed that a coherent screenwriting process to generate scenarios exists. Incorporating their ideas about the screenwriting process, it was determined that the final artifact screenplay is not the real value of the screenwriting process. The real value of the screenplay process included all of the data and processes up to, and including, the

Exploratory Draft. After a scenario is created using the Plot Storyline process, a Step Outline and Exploratory Draft process are performed. The Exploratory Draft proved to be the most important artifact because it took into account all the data that went into generating Scenario data, as well as, the Step Outline process, and Subtext data. These processes and data are the most relevant parts of the screenwriting process respective to requirements elicitation.

This is multi-disciplinary study took the complexities of separate disciplines and then interpreted how one benefitted from the other. This study positioned the screenwriting process upon the CREWS framework to interpret the screenwriting process from a Requirements Engineering (RE) perspective. In the Context facet, screenwriting scenarios provided a wider perspective by describing influence on the system from external factors. The largest perceived advantage of the screenwriting process was in capturing the Argumentation facet: (a) describing alternative solutions to a problem; (b) describing problems or conflicts themselves; and (c) displaying the choices of a particular position. In the Coverage facet, the screenwriting process allowed an understanding of the organization’s objectives, intentions and goals. In the Role facet, the screenwriting process proved to be useful when several different possible solutions for satisfying a given system requirement have to be explored. In the Operation facet, the screenwriting process provided a means to transform scenarios to make them easier to understand, more reusable, and may even add new knowledge into the scenario content.

This study verified Norden’s (2007) previously unproven claim that screenwriting techniques can be used in RE. Using the CREWS framework proved that the screenwriting process is a viable alternative to other techniques that have been evaluated by CREWS. This paper proved that screenwriting techniques complementary to the CREWS framework created advantages when building essential scenarios for software requirements elicitation.

Problems, Issues, and the Course of the Research

This research study is qualitative because it is based on the researcher's description and insight. It is based on the researcher's ability to describe screenwriting from a requirements engineer's point of view. The conclusions of this study do not suggest that there was only one correct answer. If another researcher were to perform the same study, different conclusions may result based on their experiences, backgrounds, and perceptions. These descriptions revealed the nature of screenwriting and its relationship to requirements elicitation. This study was interpretive. It allowed new insight into requirements elicitation. This paper does not show if the screenwriting process is a better scenario based approach to requirements when compared with many methods suggested over the past three decades.

Future Work

This study was an early attempt to investigate how screenwriting can be used as a scenario based approach to requirements elicitation and further research is strongly encouraged. The literature would benefit from showing how requirements elicitation scenarios are created using the screenwriting process this report outlined. A study might also be done to find out what requirements elicitation scenario techniques work best in comparison to the screenwriting process. Another avenue of future work could be creating an artifact system using screenwriting methodology. Perhaps an XML schema or some other such database repository is possible (Hobbs, 2005). This study reveals qualities of the screenwriting process that may provide advantages when building scenarios for requirements elicitation and more work needs to be done to see if these perceived advantages have value in an industrial setting.

References

Abran, A., & Moore. J. W. (2004). *Guide to the software engineering body of knowledge: 2004 edition*. Montreal: IEEE Computer Society. Retrieved from http://www.books24x7 .com/books24x7.asp

Achour, C., Ben, C., Cauvet , C., Dubois, E., Haumer, P., Heymans, P., Jarke, et al. (1996). A proposal for a scenario classification framework. *Requirements Engineering*, 4(1), 23-47. doi:10.1007/BF02802919

Achour, C. & Rolland, C. (1998, March). Guiding the construction of textual use case specifications. *Data & Knowledge Engineering Journal*, 25(1-2), 125-160. doi:10.1016/S0169-023X(97)64223-4

Ambler, S. (2005). A manager's introduction to the rational unified process. *Ambysoft Incorporated Website*. Retrieved from http://www.ambysoft.com/unifiedprocess/ rupIntroduction.html

Antón, A. I., Potts, C., & Takahashi, K. (1994A, March). Inquiry-based requirements analysis. *IEEE Software*, 11(2), 21-32. doi:10.1109/52.268952

Antón, A. I., Potts, C., & Takahashi, K. (1994B). Inquiry-based scenario analysis of systems requirements. Technical report GIT-CC-94/14, Georgia Institute of Technology. Retrieved from https://smartech.gatech.edu/bitstream/1853/6716/1/GIT-C-94-14.pdf

Alexander, I., & Maiden, N. (2004). *Scenarios, stories, use cases: Through the systems development life-cycle*. Chichester, UK: John Wiley & Sons. Available from http://www.books24x7.com/books24x7.asp

Booch, G., Jacobson, I., & Rumbaugh, J. (1999 May/June). The unified process. *IEEE Software*, 16(3), 96-102. doi:10.1109/MS.1999.10013

Carroll, J. M. (2000). *Making use: Scenario-based design of human-computer interactions*. The MIT Press. Retrieved from http://www.books24x7.com/books24x7.asp

Carroll, J. M. & Go, K. (2004, November). The blind men and the elephant: views of scenario-based system design. *Interactions*, 11(6), 44-53. doi= http://doi.acm.org/ 10.1145/1029036.1029037

Chen, C., Gao, J., Hsia, P., Kung, D., P., Samuel, J., Toyoshima, Y. (March, 1994). Formal approach to scenario analysis found. *IEEE Software*, 11(2), 33-34. doi:10.1109/52/268953

Christensen, M.J., & Thayer, R. H. (2005). *Software engineering, volume 1: The development process*. Retrieved from http://www.books24x7.com/books24x7.asp

Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM*, 31(11), 1268-1287. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.5865&rep=rep1&type=pdf

Davis, A., & Hickey, A. (2003, January). Requirements elicitation and elicitation technique selection: A model for two knowledge-intensive software development processes. *Proceedings of the thirty-sixth annual Hawaii international conference on systems sciences* (HICSS), IEEE CS. doi:10.1109/HICSS.2003.1174229

Easterbrook, S. & Nuseibeh, B. (2000). Requirements engineering: A roadmap. *Proceedings of the conference on the future of software engineering 2000: International conference on software engineering*. doi:10.11/336512.336523

Field, S. (2005). *Screenplay: The foundations of screenwriting*. New York: Bantam Dell.

Flowcharting Techniques (March,1969). White Plains, NY: International Business Machines Corporation. *Fachhochschule Jena University Website*. Retrieved from http://www.fhjena.de/~kleine/history/software/IBM-FlowchartingTechniques-GC20-8152-1.pdf

Gomaa, H. & Scott, D. B. H. (1981). Prototyping as a tool in the specification of user requirements. ICSE, editor, 5th international conference of software engineering. Available from http://portal/acm.org/portal.cfm

Grosz, G., Kla, R., & Rolland, C. (1999). Experience with goal-scenario coupling in requirements engineering. CREWS Report 98-32. doi:10.1109/ISRE.1999.777987

Hertzum, M. (2003, February). Making use of scenarios: A field study of conceptual design. *International Journal of Human-Computer Studies*, 58, 215-239. doi:10.1016/S1071-5819(02)00138-6

Hobbs, R., & Potts, C. (2000, April). Hyperscenarios: A framework for active narrative. Proceedings of the 38th Annual on Southeast Regional Conference (Clemson, South Carolina, April 07 - 08, 2000). doi:10.1145/1127716.1127760

Hobbs, R. (2005, August). A scenario-directed computational framework to aid decision-making and systems development. Retrieved from http://citeseerx.ist.psu.edu/ viewdoc/download?doi=10.1.1.92.9043&rep=rep1&type=pdf

Holbrook, H. (1990, January). A scenario-based methodology for conducting requirements elicitation. *SIGSOFT Software Engineering Notes*, 15(1), 95-104. doi=10.1145/382294.382725

Hooper, J. W. & Hsia, P. (1982). Scenario-based prototyping for requirements identification. In Proceedings of the Workshop on Rapid Prototyping (Columbia, Maryland, April 19 - 21, 1982), 88-93. doi:10.1145/382294.382725

IEEE (1990). *IEEE Std 610.12-1990: IEEE standard glossary of software engineering* *terminology*. Los Alamitos, CA: IEEE Computer Society Press. Retrieved from http://pswlab.kaist.ac.kr/courses/cs550spring09/IEEE_standard _terminology.pdf

Kyng, M. (1995, September). Making representations work. *Communications of the ACM*, 38 (9), 46-55. doi:10.1145/223248.223261

Linberg, K.R. (1999). Software development perceptions about software project failure: A case study. *The Journal of Systems and Software, 49 (12), 177–192*. Retrieved from http://tao.nuk.edu.tw/TW/reference/softwareDeveloper.pdf

McKee, R. (1997). *Story: Substance, structure, style, and the principles of screenwriting*. New York: HarperCollins Publishers Inc.

Nielson, L. (2002, June). From user to character: An investigation into user-descriptions in scenarios. *Proceedings of the 4th conference on designing interactive systems: Processes, practices, methods, and techniques*. doi:10.1145/778712.778729

Norden, B. (2007 July/August). Screenwriting for requirements engineers. *IEEE Software,* 24(4). doi:10.1109/MS.2007.117

Pohl, K. (1997). *Requirements engineering: An overview, encyclopedia of computer science and technology* (Vol. 36). New York: Marcel Dekker, Inc. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.67.1394&rep= rep1&type=pdf.

Potts, C. (1995, August). Using schematic scenarios to understand user needs. *Proceedings of the 1st conference on designing interactive systems: Processes, practices, methods, & techniques* (Ann Arbor, Michigan, United States, August 23 - 25, 1995). doi:10.1145/225434.225462

Royce, W. (1970, August). Managing the development of large software systems: Concepts and techniques. *Proceedings of WESCON*. Retrieved from http://leadinganswers.typepad.com /leading_answers/files/original_waterfall_paper_winston_royce.pdf

Rinzler, B. (2009) *Telling stories: A short path to writing better software requirements*. Indianapolis: John Wiley & Sons. Retrieved from http://www.books24x7.com /books24x7.asp

Rubenstein, D. (2007, March). Standish group report: There's less development chaos today. *Software Development Times*. Retrieved from http://www.sdtimes.com/ article/story-20070301-01.html

Sawyer, P., & Sommerville, I. (1997). *Requirements engineering: A good practice guide.* West Sussex: John Wiley & Sons. Retrieved from http://www.books24x7.com/books24x7.asp

Sawyer, P., Sommerville, I., & Viller, S. (1997). Viewpoints: Principles, problems and a practical approach to requirements engineering. *Annals of Software Engineering*, 3(1), 101-130. doi:10.1023/A:1018946223345

Sumner, M. (1999). Critical success factors in enterprise wide information management systems projects. *Proceedings of the Americas Conference on Information Systems (AMCIS)*, 232-234. Retrieved from http://www.bus.ucf.edu/shornik/ism6121/Readings /enterprise_riskfactors.pdf

Wiegers, K. E. (2003). *Software requirements.* Redmond, WA: Microsoft Press. Retrieved from http://www.books24x7.com/books24x7.asp