

Chapter 3 Case Studies

Case One: The NextGen POS System

- The NextGen POS System
 - A POS system is an application used to record sales, handle payments. It is typically used in retail stores. It includes hardware components such as bar code scanner, terminal...
 - interfaces third party systems (tax, inventory control systems).

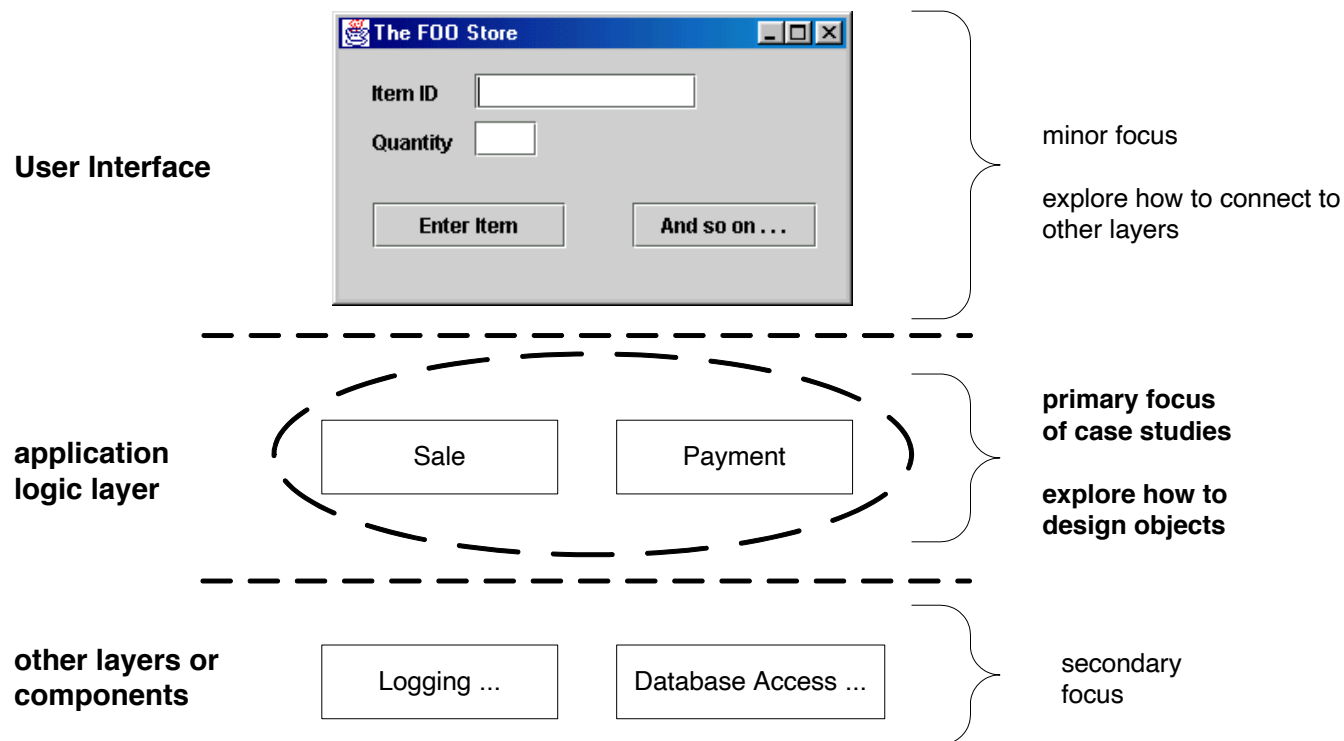


Case Two: Monopoly (very simplified)

- To show OOA/D can apply to very different problems a boardgame is the second case study.
- The game will run as a simulation
- One person will start the game, and indicate the number of simulated players, and then watch while the runs to completion.
- Although not as complicated as NextPos, it is rich enough for domain modeling, design, applying patterns.



Although OO technology can be applied at all levels, in this introduction to OOA/D focuses on the *core application logic layer*, with some secondary discussion of the other layers.



Although OO technology can be applied at all levels, in this introduction to OOA/D focuses on the *core application logic layer*, with some secondary discussion of the other layers.

- Other layers are usually very technology/platform dependent. For example, to explore the OO design of a Web UI or rich client UI layer in Java, we would need to learn in detail about a framework such as Struts or Swing. But for .NET or Python, the choice and details are very different.
- In contrast, the OO design of the core logic layer is similar across technologies.
- The essential OO design skills learned in the context of the application logic layer are applicable to all other layers or components.
- The design approach/patterns for the other layers tends to change quickly as new frameworks or technologies emerge. For example, in the mid-1990s developers would probably build their own home-grown object-relational database access layer. Some years later, they were more likely to use a free, open-source solution such as Hibernate (if Java technology).

Case Study Strategy: Iterative Development+Iterative Learning



Iteration 1

Introduces just those analysis and design skills related to iteration one.



Iteration 2

Additional analysis and design skills introduced.

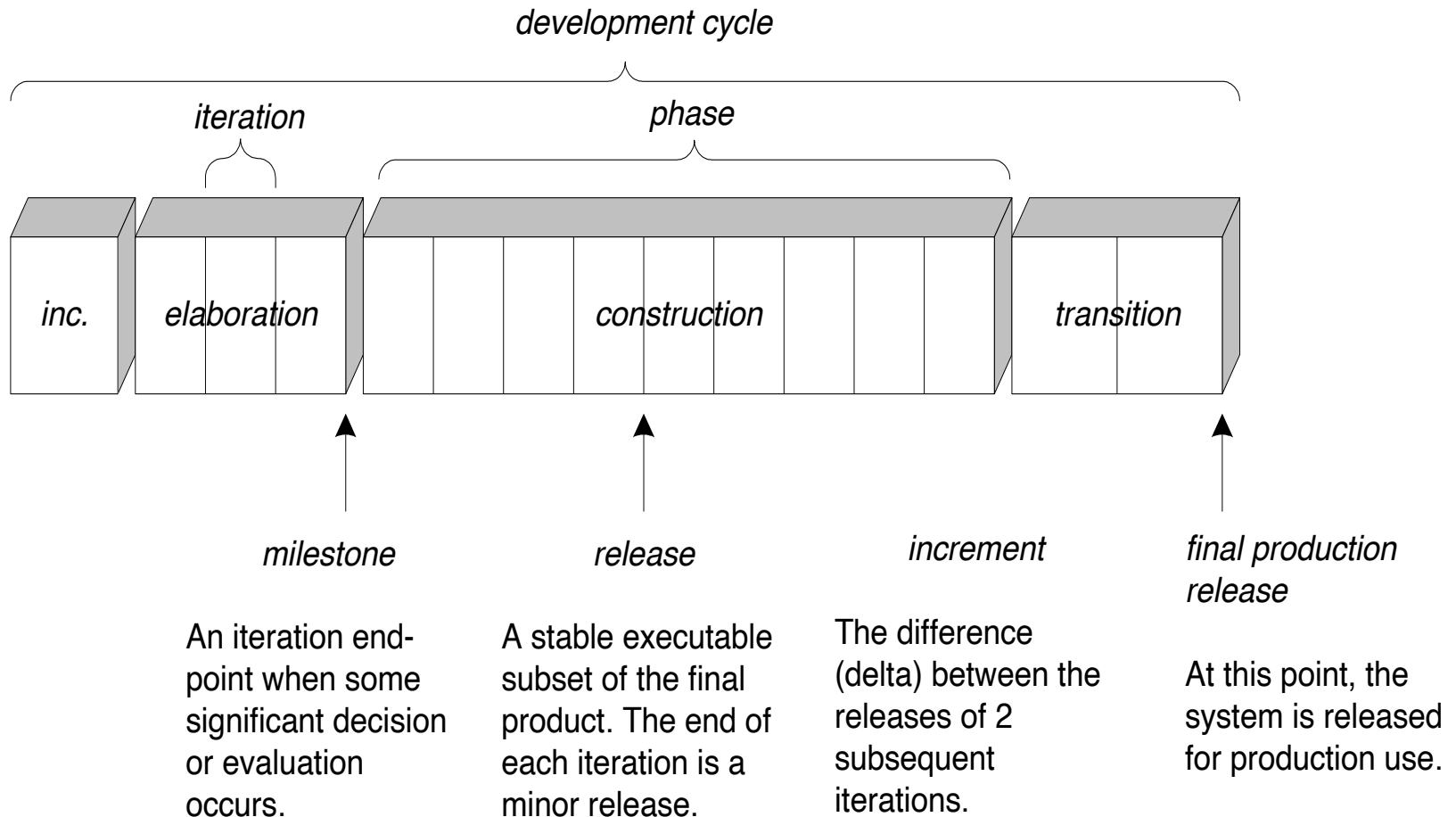


Iteration 3

Likewise.

PART 2

Inception Phase



The Inception Phase

- Inception: Short initial phase
 - What is the vision for this project?
 - Is it feasible?
 - Rough estimate of cost
 - Should we proceed or go?
- Do some preliminary requirements and use cases work
 - It will include analysis of perhaps 10% of the requirements (most requirement analysis occurs during the elaboration phase)
- In your class project, we will assume that this phase has been already carried out
 - Earlier work tells you that the project is feasible
 - Preliminary requirements have been explored

This is a critical point, and repeatedly misunderstood on UP projects when people superimpose old "waterfall" thinking. The UP is not the waterfall, and the first phase, inception, is not the time to do all requirements or create believable estimates or plans. That happens during elaboration.

What Artifacts may start in Inception

Artifact ^[†]	Comment
Vision and Business Case	Describes the high-level goals and constraints, the business case, and provides an executive summary.
Use-Case Model	Describes the functional requirements. During inception, the names of most use cases will be identified, and perhaps 10% of the use cases will be analyzed in detail.
Supplementary Specification	Describes other requirements, mostly non-functional. During inception, it is useful to have some idea of the key non-functional requirements that have will have a major impact on the architecture.
Glossary	Key domain terminology, and data dictionary.
Risk List & Risk Management Plan	Describes the risks (business, technical, resource, schedule) and ideas for their mitigation or response.
Prototypes and proof-of-concepts	To clarify the vision, and validate technical ideas.
Iteration Plan	Describes what to do in the first elaboration iteration.
Phase Plan & Software Development Plan	Low-precision guess for elaboration phase duration and effort. Tools, people, education, and other resources.
Development Case	A description of the customized UP steps and artifacts for this project. In the UP, one always customizes it for the project.

Some artifacts will be partially completed in this phase and iteratively refined in subsequent iterations

The purpose of inception is to collect just enough information to establish a common vision, decide if moving forward is feasible

Most UML diagramming will occur in elaboration phase

Vision

Revision History

Version	Date	Description	Author
inception draft	Jan 10, 2031	First draft. To be refined primarily during elaboration.	Craig Larman

Introduction

We envision a next generation fault-tolerant point-of-sale (POS) application, NextGen POS, with the flexibility to support varying customer business rules, multiple terminal and user interface mechanisms, and integration with multiple third-party supporting systems.

The analysis in this example is illustrative, but fictitious.

Positioning

Business Opportunity

Existing POS products are not adaptable to the customer's business, in terms of varying business rules and varying network designs (for example, thin client or not; 2, 3, or 4-tier architectures). In addition, they do not scale well as terminals and business increase. And, none can work in either on-line or off-line mode, dynamically adapting depending on failures. None easily integrate with many third-party systems. None allow for new terminal technologies such as mobile PDAs. There is marketplace dissatisfaction with this inflexible state of affairs, and demand for a POS that rectifies this.

Problem Statement

Traditional POS systems are inflexible, fault intolerant, and difficult to integrate with third-party systems. This leads to problems in timely sales processing, instituting improved processes that don't match the software, and accurate and timely accounting and inventory data to support measurement and planning, among other concerns. This affects cashiers, store managers, system administrators, and corporate management.

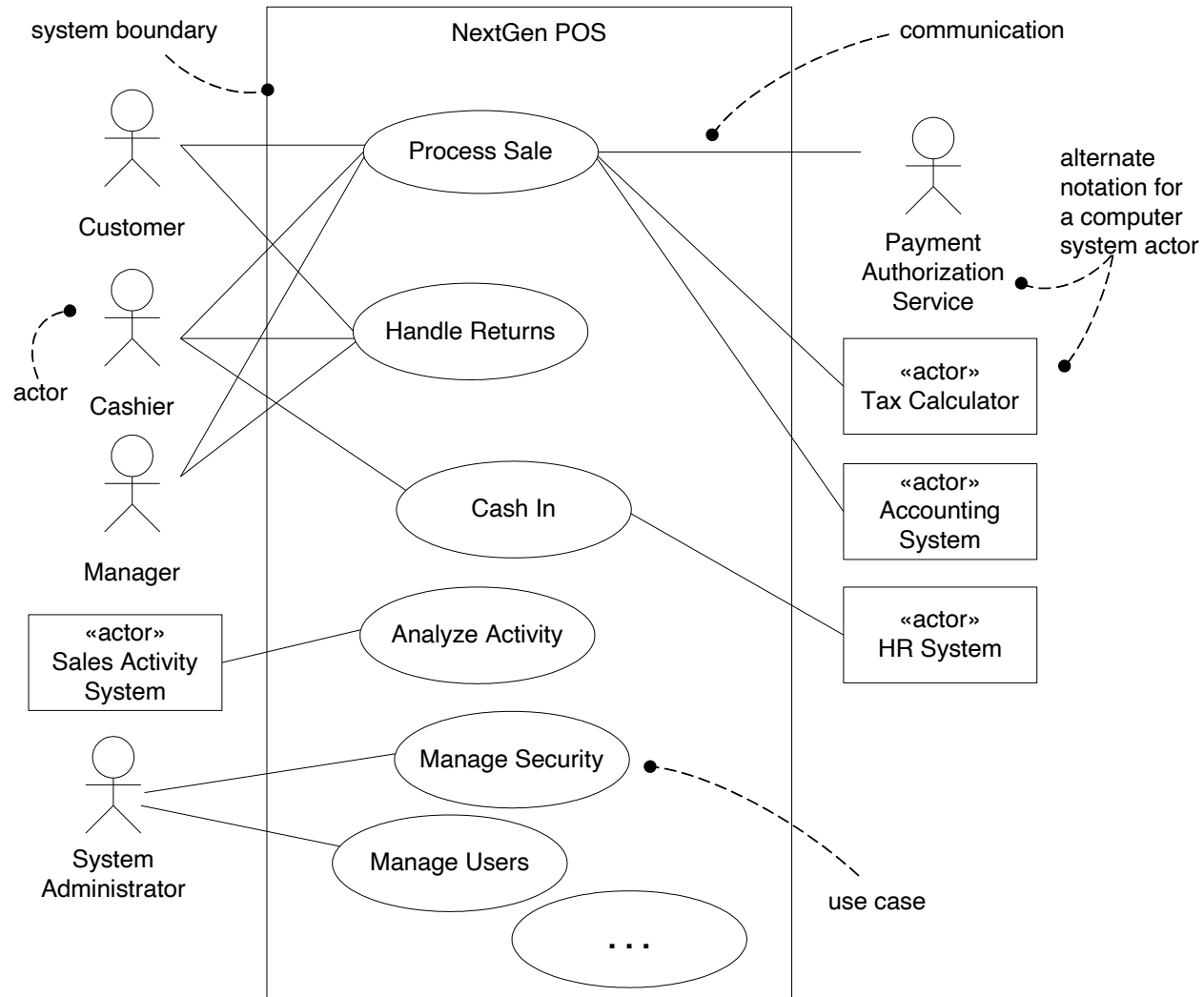
Product Position Statement

Terse summary of who the system is for, its outstanding features, and what differentiates it from the competition.

Alternatives and Competition...

Stakeholder Descriptions

Understand who the players are, and their problems.



Glossary

Revision History

Version	Date	Description	Author
Inception draft	Jan 10, 2031	First draft. To be refined primarily during elaboration.	Craig Larman

Definitions

Term	Definition and Information	Format	Validation Rules	Aliases
item	A product or service for sale			
payment authorization	Validation by an external payment authorization service that they will make or guarantee the payment to the seller.			
payment authorization request	A composite of elements electronically sent to an authorization service, usually as a char array. Elements include: store ID, customer account number, amount, and timestamp.			
UPC	Numeric code that identifies a product. Usually symbolized with a bar code placed on products. See www.uc-council.org for details of format and validation.	12-digit code of several subparts.	Digit 12 i s a check digit.	Universal Product Code
...	...			

Supplementary Specification

Revision History

Version	Date	Description	Author
Inception draft	Jan 10, 2031	First draft. To be refined primarily during elaboration.	Craig Larman

Introduction

This document is the repository of all NextGen POS requirements not captured in the use cases.

Functionality

(Functionality common across many use cases)

Logging and Error Handling

Log all errors to persistent storage.

Pluggable Rules

At various scenario points of several use cases (to be defined) support the ability to customize the functionality of the system with a set of arbitrary rules that execute at that point or event.

Security

All usage requires user authentication.

Usability

Human Factors

The customer will be able to see a large-monitor display of the POS. Therefore:

- Text should be easily visible from 1 meter.
- Avoid colors associated with common forms of color blindness.

Speed, ease, and error-free processing are paramount in sales processing, as the buyer wishes to leave quickly, or they perceive the purchasing experience (and seller) as less positive.

The cashier is often looking at the customer or items, not the computer display. Therefore, signals and warnings should be conveyed with sound rather than only via graphics.

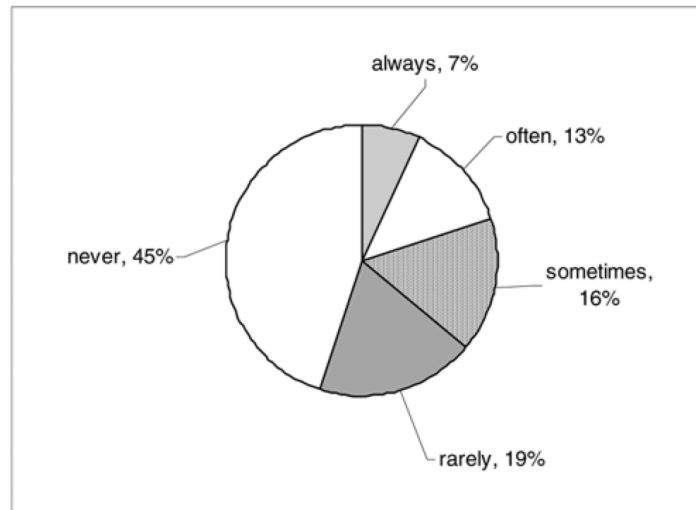
Chapter 5 Evolutionary Requirements

- Definition: **Requirements** are the capabilities that the system must support and the conditions to which it must conform.
- RUP approach to requirements
 - A systematic approach to
 - finding,
 - documenting,
 - organizing, and
 - tracking**the changing requirements of a system**
 - Doing it iteratively

Evolutionary vs Waterfall Requirements

- In UP and other evolutionary methods (Scrum, XP...), start production-quality programming and testing long before most of the requirements have been analyzed, when only 10% or 20% of the most significant, risky, high-business value requirements have been specified.
- When waterfall requirements analysis is attempted, most early specified features were of little-of-use.

Figure 5.1. Actual use of waterfall-specified features.



UP Categories of requirements: FURPS+

- FURPS
 - **Functional**: Features, capabilities, security
 - **Usability**: Human factors, help, documentation
 - **Reliability**: Frequency of failure, recoverability, predictability
 - **Performance**: Response times, throughput, accuracy, availability, resource usage
 - **Supportability**: Adaptability, maintainability, internationalization, configurability
- +
 - **Implementation**: Resource limitations, languages and tools, hardware
 - **Interface**: Interfacing with external systems
 - **Operations**: System management where it is going to be used
 - **Packaging**
 - **Legal**

How are requirements organized and stored?

- UP artifacts
 - Use-case model: A set of typical scenarios of using a system
 - Supplementary specification: All non-functional requirements, features
 - Glossary
 - Key terms
 - Data dictionary
 - Vision
 - Big ideas of project
 - The business case
 - Business rules: Rules of the domain you have to operate in
 - Tax laws
 - Aviation laws
 - Laws of physics