Table of Contents

Domain analysis and modeling

> He who asks a question remains a fool for five minutes. He who does not ask remains a fool forever.
>
> — Chinese Proverb

Introduction

As we saw in the previous chapter, misinterpreted requirements cause a significant portion of software projects to fail. Arriving at a shared understanding and creating a useful domain model necessitates high degrees of collaboration with domain experts. In this chapter, we will introduce the sample application we will use throughout the book and explore modeling techniques such as domain storytelling and eventstorming to enhance our collective understanding of the problem in a reliable and structured manner.

The following topics will be covered in this chapter:

- Introducing the example application (Letter of Credit)
- Enhancing shared understanding
- Domain storytelling
- EventStorming

This chapter will help developers and architects learn how to apply these techniques in real-life

situations to produce elegant software solutions that mirror the problem domain that needs to be solved. Similarly, non-technical domain experts will understand how to communicate their ideas and collaborate effectively with technical team members to accelerate the process of arriving at a shared understanding.

Technical requirements

There are no specific technical requirements for this chapter. However, given that it may become necessary to collaborate remotely as opposed to being in the same room with access to a whiteboard, it will be useful to have access to the following:

1. Digital whiteboard (like https://www.mural.co/ or http://miro.com/)
2. Online domain storytelling modeler (like https://www.wps.de/modeler/)

Understanding Letter of Credit (LC)

Documentary Letter of Credit (LC) is a financial instrument issued by the banks as a contract between the importer (or buyer) and the exporter (or seller). This contract specifies terms and conditions of the transaction under which importer promises to pay the exporter in exchange for the goods or services provided by the exporter. Letter of Credit transaction typically involves multiple parties. A simplified summary of the parties involved is described below:

1. **Importer**: The buyer of the goods or services.
2. **Exporter**: The seller of the goods or services.
3. **Freight Forwarder**: The agency that handles shipment of goods on behalf of the exporter. This is only applicable in cases there is an exchange of physical goods.
4. **Issuing Bank**: The bank that the importer requests to issue the LC application. Usually the importer has a pre-existing relationship with this bank.
5. **Advising Bank**: The bank that informs the exporter about the issuance of the LC. This is usually a bank that is native to the exporter's country.
6. **Negotiating Bank**: The bank that the exporter submits documents for the shipment of goods, or the services provided. Usually the exporter has a pre-existing relationship with this bank.
7. **Reimbursement Bank**: The bank that reimburses the funds to the negotiating bank, at the request of the issuing bank.

It is important to note that the same bank can play more than one role for a given transaction. In the most complex cases, there can be four distinct banks involved for a transaction (sometimes even more, but we will skip those cases for brevity).

The LC issuance application

As discovered in the previous chapter, Kosmo Primo Bank needs us to focus on streamlining the process used for LC application and issuance functions. In this chapter, and indeed the rest of this book, we will strive to understand, evolve, design and build a software solution to make the process more efficient by replacing the largely manual and error-prone workflows with more simplified

processes based on larger amounts of automation.

We understand that unless one is an expert dealing with international trade, it is unlikely that one would have an intimate understanding of concepts like Letters of Credit (LCs). In the upcoming section, we will look at demystifying LCs and how to work with them.

Enhancing shared understanding

When working with a problem where domain concepts are unclear, there is a need to arrive at a common understanding among key team members (both those that have bright ideas — the business/product people, and those that translate those ideas into working software — the software developers). For this process to be effective, we tend to look for approaches that are:

- Quick, informal and effective
- Collaborative - Easy to learn and adopt for both non-technical and technical team members
- Pictorial - because a picture can be worth a thousand words
- Usable for both coarse grained and fine-grained scenarios

There are several means to arrive at this shared understanding. Some commonly used approaches are listed below:

- UML
- BPMN
- Use Cases
- User Story Mapping
- CRC Models
- Data Flow Diagrams

The above-mentioned modeling techniques have tried to formalize knowledge and express them in form of a structure diagram or text to help in delivering the business requirements as a software product. However, this attempt has not narrowed but has widened the gap between the business and the software systems. While these methods tend to work well for technical audiences, they are usually not as appealing to non-technical users.

In order to restore the balance and promote the use of techniques that can work for both parties, we will use **domain storytelling** and **eventstorming** as our means to capture business knowledge from domain experts for consumption of developers, business analysts etc.

Domain storytelling

> You’re never going to kill storytelling because it’s built into the human plan. We come with it.
>
> — Margaret Atwood

Introducing Domain Storytelling

Scientific research has now proven that learning methods that employ audiovisual aids assist both the teacher and the learners in retaining and internalizing concepts very effectively. In addition, teaching what one has learnt to someone else helps reinforce ideas and also stimulates the formation of new ones. Domain storytelling is a collaborative modeling technique that combines a pictorial language, real-world examples, and a workshop format to serve as a very simple, quick and effective technique for sharing knowledge among team members. Domain Storytelling is a technique invented and popularized by Stefan Hofer and Henning Schwentner based on some related work done at the University of Hamburg called *cooperation pictures*.

A pictorial notation of the technique is illustrated in the diagram below:

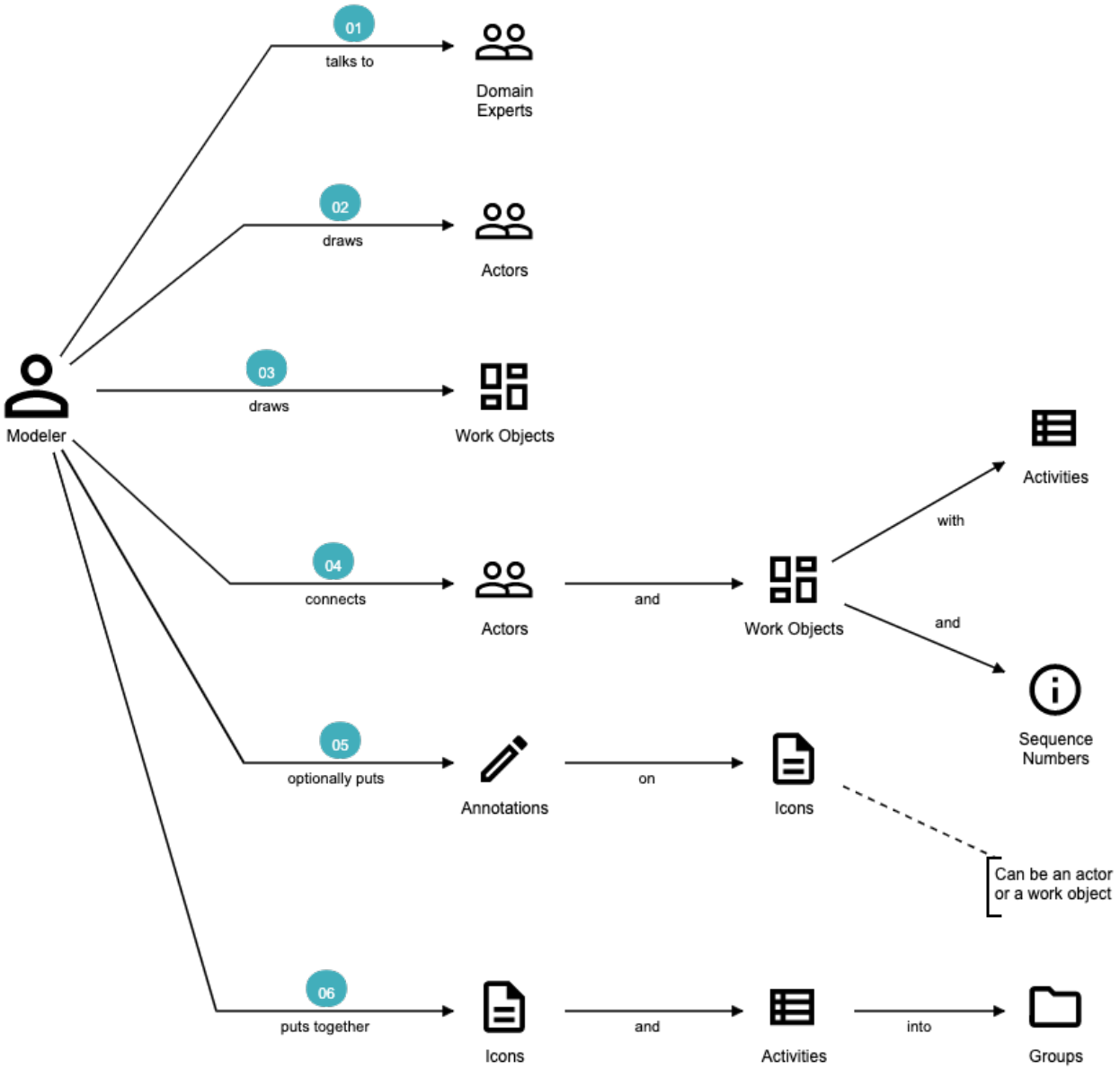

Figure 1. Domain storytelling summarized

A domain story is conveyed using the following attributes:

Actors - Stories are communicated from the perspective of an actor (noun), for example, the issuing bank, who plays an active role in the context of that particular story. It is a good practice to use the

ubiquitous language for the particular domain.

Work Objects - Actors act on some object, for example, applying for an LC. Again, this would be a term (noun) commonly used in the domain.

Activities - Actions (verb) performed by the actor on a work object. Represented by a labelled arrow connecting the actor and the work object.

Annotations - Used to capture additional information as part of the story, usually represented in few sentences.

Sequence Numbers - Usually, stories are told one sentence after the other. Sequence numbers helps capture the sequence of the activities in a story.

Groups - An outline to represent a collection of related concepts ranging from repeated/optional activities to subdomains/organizational boundaries.

Using DST for the LC application

XYZ Bank has a process that allows processing of LCs. However, this process is very archaic, paper-based and manually intensive. Very few at the bank fully understand the process end-to-end and natural attrition has meant that the process is overly complex without good reason. So they are looking to digitize and simplify this process. DST itself is just a graphical notation which can be done in isolation. However, it is typical to not do this on your own and employ a workshop style with domain experts and software experts working collaboratively.

In this section, we will employ a DST workshop to capture the current business flow. The following is an excerpt of such a conversation between **Katie**, *the domain expert* and **Patrick**, *the software developer*.

Patrick : *"Can you give me a high level overview of a typical LC Flow?"*
Katie : *"Sure, it all begins with the importer and the exporter entering into a contract for purchase of goods or services."*
Patrick : *"What form does this contract take? Is it a formal documentClause? Or is this just a conversation?"*
Katie : *"This is just a conversation."*
Patrick : *"Oh okay. What does the conversation cover?"*
Katie : *Several things — nature and quantity of goods, pricing details, payment terms, shipment costs and timelines, insurance, warranty, etc. These details may be captured in a purchase order — which is a simple documentClause elaborating the above.*

At this time, Patrick draws this part of the interaction between the importer and the exporter. This graphic is depicted in the following diagram:

Figure 2. Interaction between importer and exporter

Patrick : *"Seems straight forward, so where does the bank come into the picture?"*
Katie : *"This is international trade and both the importer and the exporter need to mitigate the financial risk involved in such business transactions. So they involve a bank as a trusted mediator."*
Patrick : *"What kind of bank is this?"*
Katie : "*Usually, there are multiple banks involved. But it all starts with an **issuing bank**.*"
Patrick : *"What is an issuing bank?"*
Katie : *"Any bank that is authorized to mediate international trade deals. This has to be a bank in the importer's country."*
Patrick : *"Does the importer need to have an existing relationship with this bank?"*
Katie : *"Not necessarily. There may be other banks with whom the importer may have a relationship with — which in turn liaises with the issuing bank on the importer's behalf. But to keep it simple, let's assume that the importer has an existing relationship with the issuing bank — which is our bank in this case."*
Patrick : *"Does the importer provide details of the purchase order to the issuing bank to get started?"*
Katie : *"Yes. The importer provides the details of the transaction by making an **LC application**."*

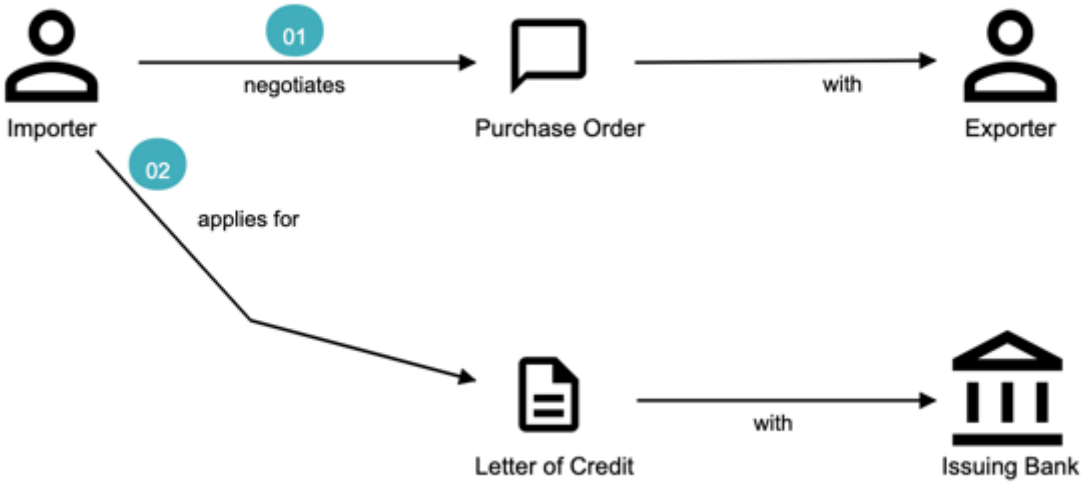

Figure 3. Introducing the LC and the issuing bank

Patrick : *"What does the issuing bank do when they receive this LC application?"*

Katie : *"Mainly two things — whet the financial standing of the importer and the legality of the goods being imported."*
Patrick : "Okay. What happens if everything checks out?"
Katie : *"The issuing bank approves the LC and notifies the importer."*

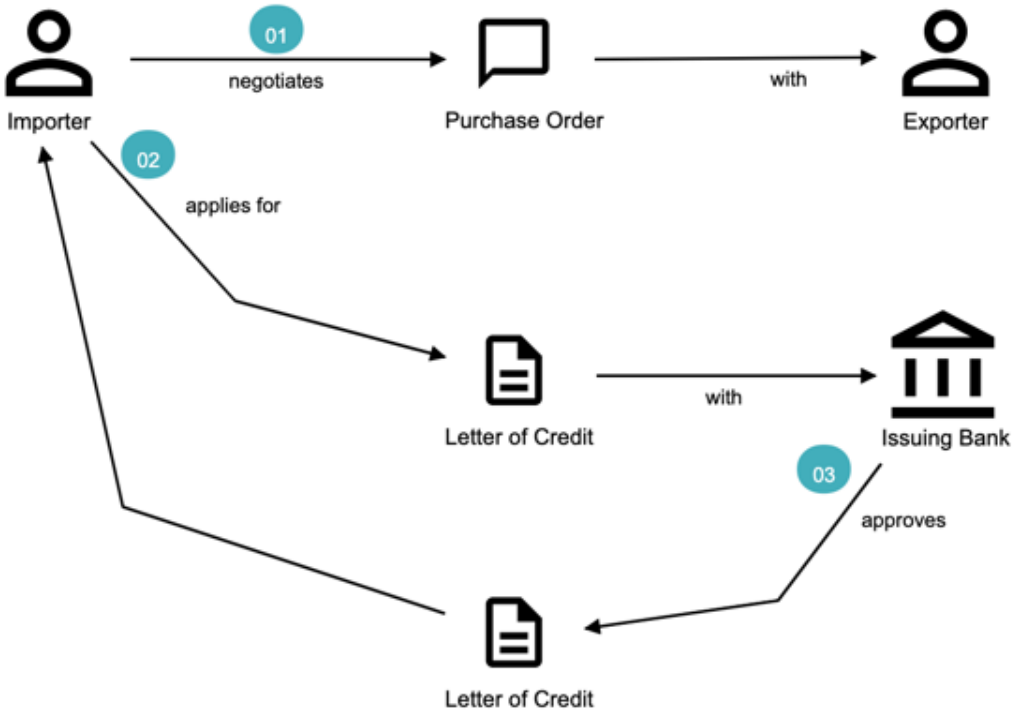

Figure 4. Notifying LC approval to the importer

Patrick : *"What happens next? Does the issuing bank contact the exporter now?"*
Katie : *"Not yet. It is not that simple. The issuing bank can only deal with a counterpart bank in the exporter's country. This bank is called the **advising bank**."*

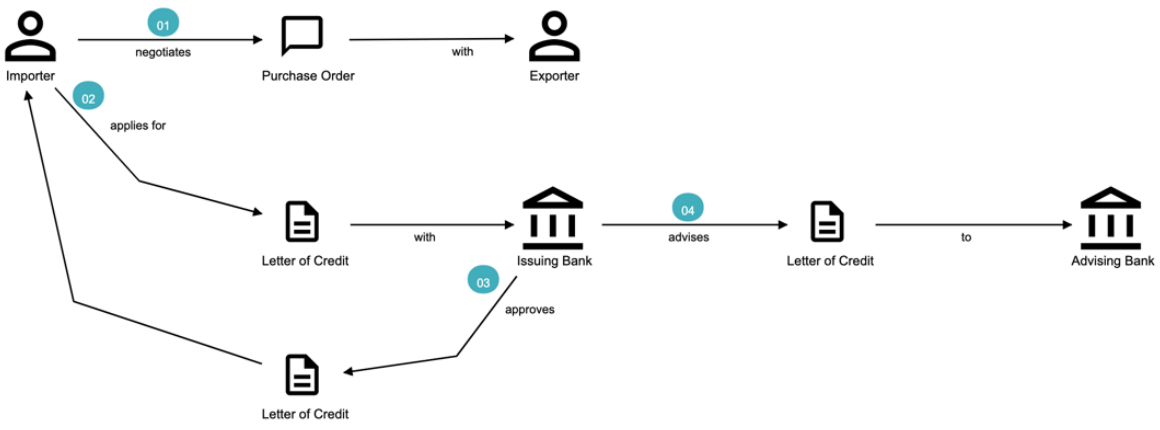

Figure 5. Introducing the advising bank

Patrick : *"What does the advising bank do?"*
Katie : *"The advising bank notifies the exporter about the LC."*
Patrick : *"Doesn't the importer need to know that the LC has been advised?"*
Katie : *"Yes. The issuing bank notifies the importer that the LC has been advised to the exporter."*

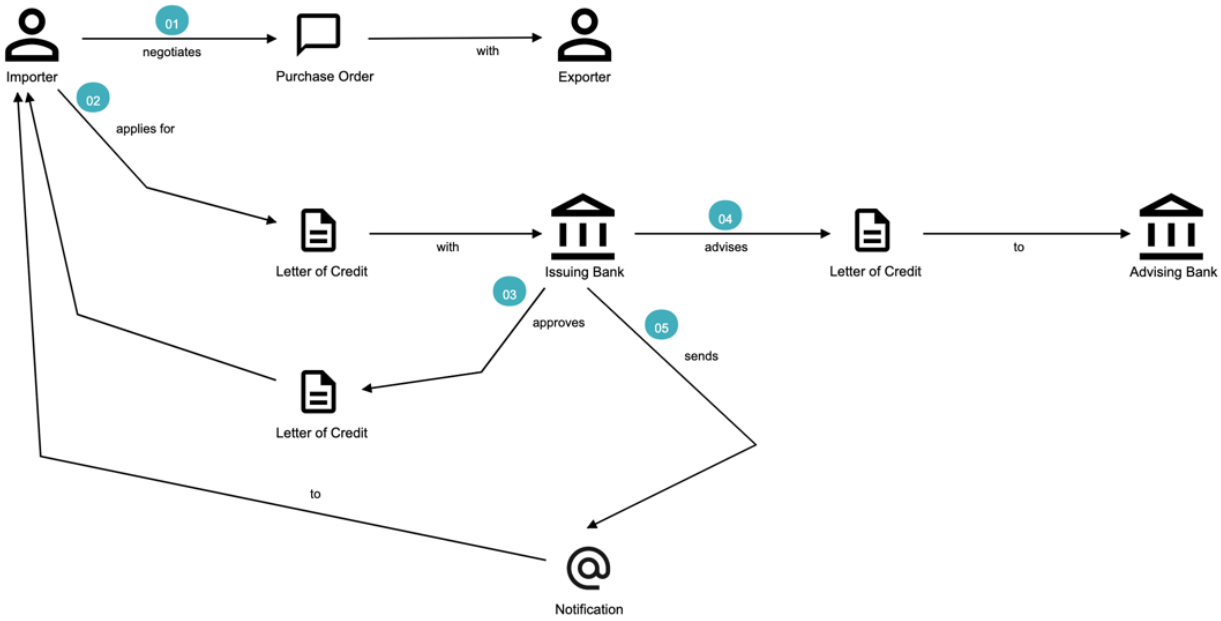

Figure 6. Advice notification to the importer

Patrick : *"How does the exporter know how to proceed?"*
Katie : *"Through the advising bank — they notify the exporter that the LC was issued."*

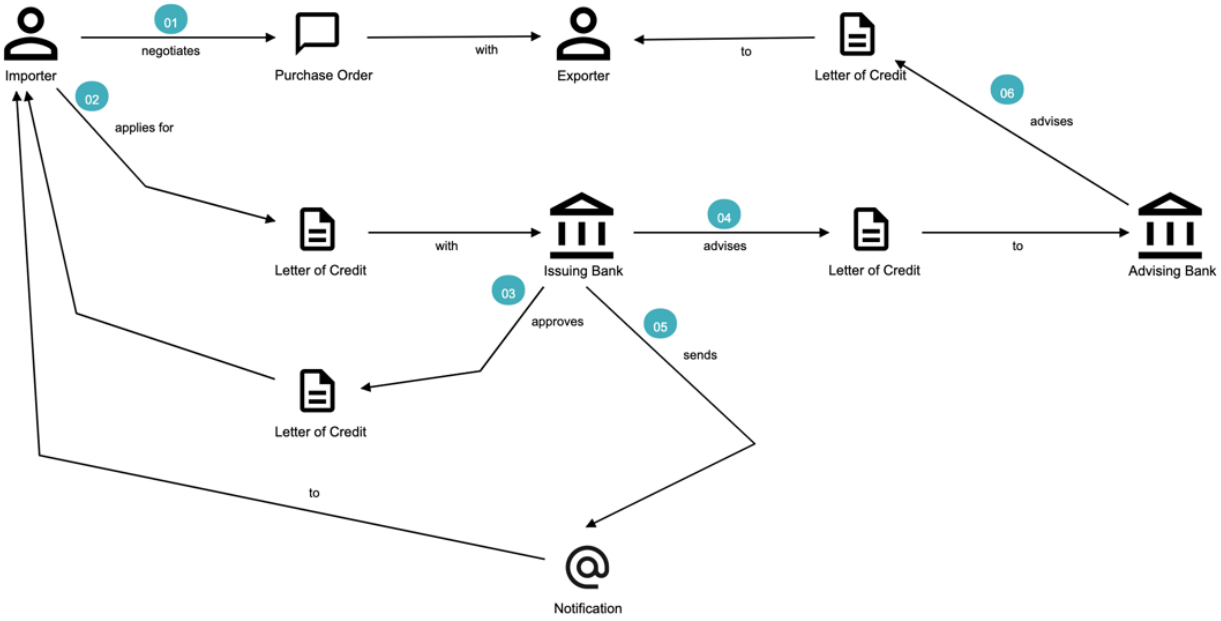

Figure 7. Dispatching the advice to the exporter

Patrick : *"Does the exporter initiate shipping at this time and how do they get paid?"*
Katie : *"Through the advising bank — they notify the exporter that the LC was issued and this triggers the next steps in the process — this process of settling the payment is called* ***settlement****. But let's focus*

on issuance right now. We will discuss settlement at a later time."

We have now looked at an excerpt of a typical DST workshop. The DST workshop has served to provide a reasonably good understanding of the high level business flow. Note that we have not referenced any technical artifacts during the process.

To be able to refine this flow and convert it into a form that can be used to design the software solution, we will need to further enhance this view. In the upcoming section, we will use EventStorming as a structured approach to achieve that.

EventStorming

> The amount of energy necessary to refute bullshit is an order of magnitude bigger than to produce it.
>
> — Alberto Brandolini

Introducing EventStorming

In the previous section, we gained a high level understanding of the LC Issuance process. To be able to build a real-world application, it will help to use a method that delves into the next level of detail. EventStorming, originally conceived by Alberto Brandolini, is one such method for the collaborative exploration of complex domains.

In this method, one simply starts by listing out all the events that are significant to the business domain in roughly chronological order on a wall or whiteboard using a bunch of colored sticky notes. Each of the note types (denoted by a color) serve a specific purpose as outlined below:

- **Domain Event**: An event that is significant to the business process — expressed in past tense.
- **Command**: An action or an activity that may result in one or more domain events occurring. This is either user initiated or system initiated, in response to a domain event.
- **User**: A person who performs a business action/activity.
- **Policy**: A set of business invariants (rules) that need to be adhered to, for an action/activity to be successfully performed.
- **Query/Read Model**: A piece of information required to perform an action/activity.
- **External System**: A system significant to the business process, but out of scope in the current context.
- **Hotspot**: Point of contention within the system that is likely confusing and/or puzzling beyond a small subsection of the team.
- **Aggregate**: An object graph whose state changes consistently and atomically. This is consistent with the definition of aggregates we saw in Chapter 2.

The depiction of the stickies for our EventStorming workshop is shown here:

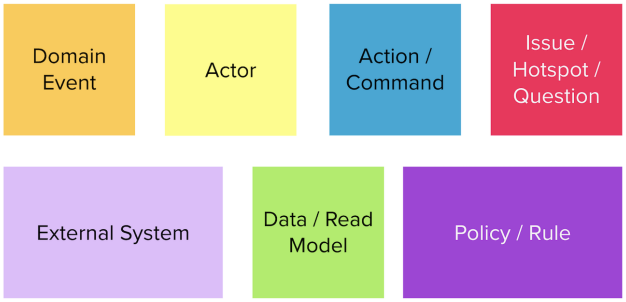

Figure 8. EventStorming legend

Why domain events? When trying to understand a business process, it is convenient to express significant facts or things that happen in that context. It can also be informal and easy for audiences that are uninitiated with this practice. This provides an easy to digest visual representation of the domain complexity.

Using eventStorming for the LC issuance application

Now that we have a high level understanding of the current business process, thanks to the domain storytelling workshop, let's look at how we can delve deeper using eventstorming. The following is an excerpt of the stages from an eventstorming workshop for the same application.

1. Outline the event chronology

During this exercise, we recall significant **domain events** (using orange stickies) that happen in the system and paste them on the whiteboard, as depicted below. We ensure that the event stickies are pasted roughly in the chronological order of occurrence. As the timeline is enforced, the business flow will begin to emerge.

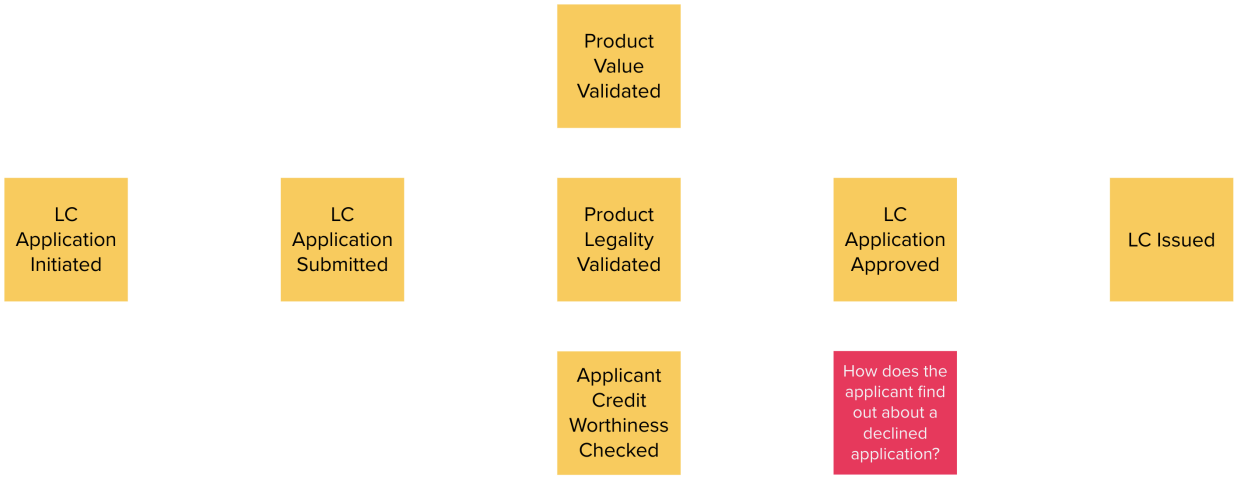

Figure 9. Event chronology

This acts as an aid to understand the big picture. This also enables people in the room to identify hotspots in the existing business process. In the above illustration, we realized that, the process to handle "declined LC applications" is sub-optimal, i.e. applicants do not receive any information when their application is declined.

To address this, we added a new domain event which explicitly indicates that an application is

declined, as depicted below:

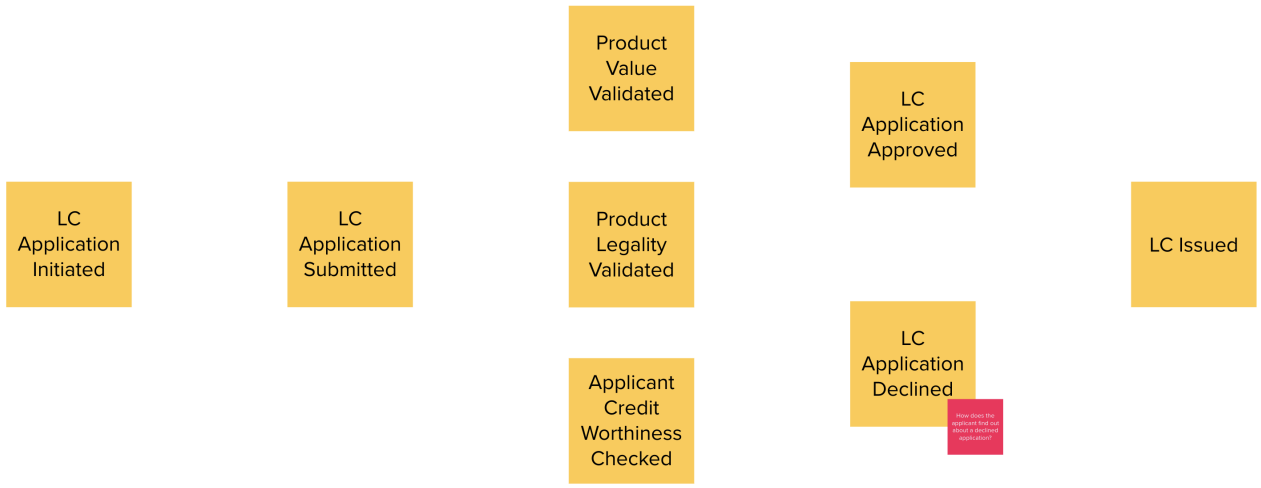

Figure 10. New event to handle declined applications

2. Identify triggering activities and external systems

Having arrived at a high level understanding of event chronology, the next step is to embellish the visual with **activities/actions** that cause these events to occur (using blue stickies) and interactions with **external systems** (using pink stickies).

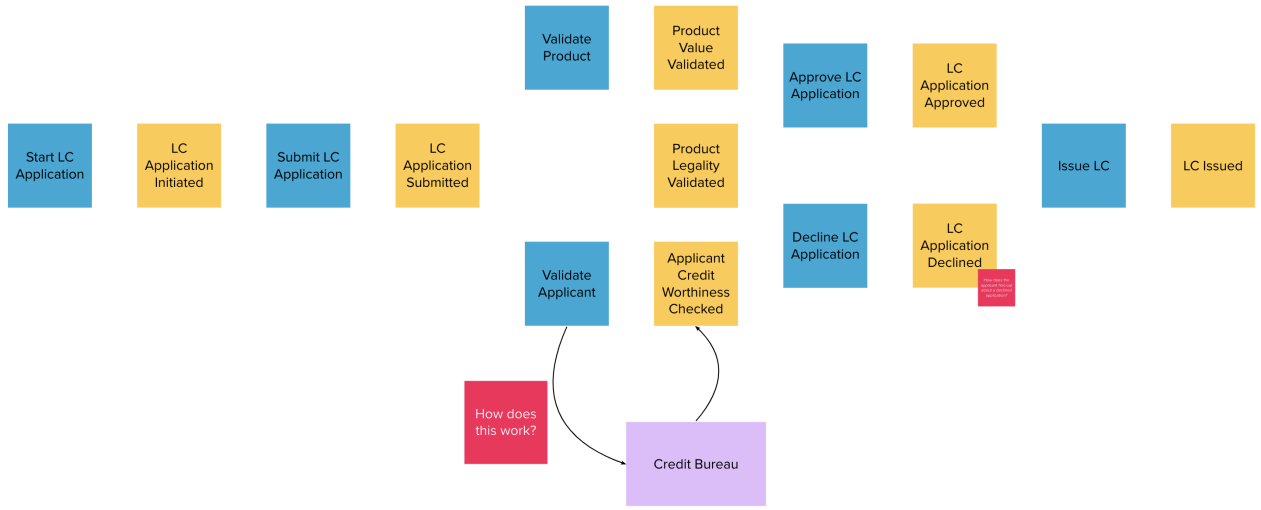

Figure 11. Activities and external systems

3. Capture users, context and policies

The next step is to capture **users** who perform these activities along with their functional **context** (using yellow stickies) and policies (using purple stickies).

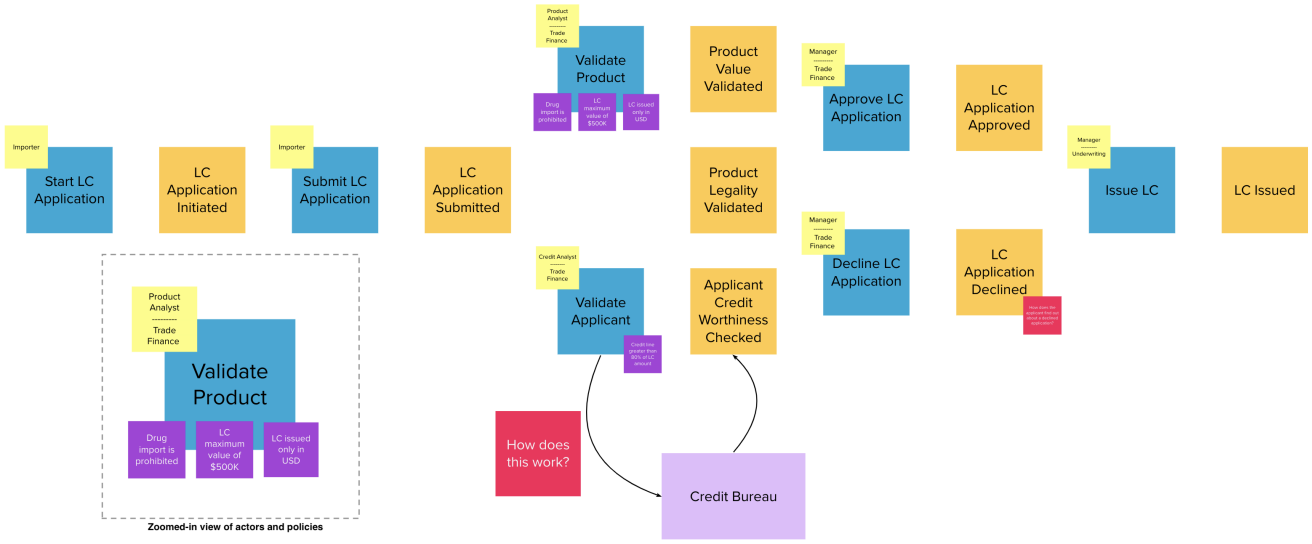

Figure 12. Users and policies

4. Outline query models

Every activity requires a certain set of data to be able to be performed. Users will need to view out-of-band data that they need to act upon and also see the result of their actions. These sets of data are represented as **query models** (using green stickies).

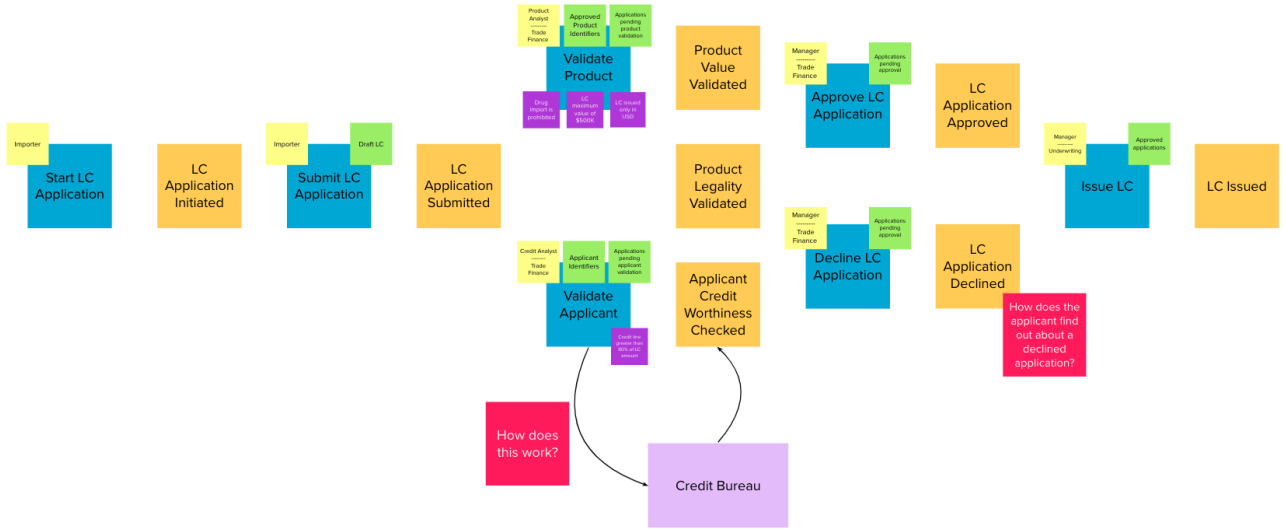

Figure 13. Big picture eventstorming workshop board

For both the domain storytelling and eventstorming workshops, it works best when we have approximately 6-8 people participating with a right mix of domain and technology experts.

This concludes the eventstorming workshop to gain a reasonably detailed understanding of the LC application and issuance process. Does this mean that we have concluded the domain requirements gathering process? Not at all — while we have made significant strides in understanding the domain, there is still a long way to go. The process of elaborating domain requirements is perpetual. Where are we in this continuum? The picture below is an attempt to clarify:

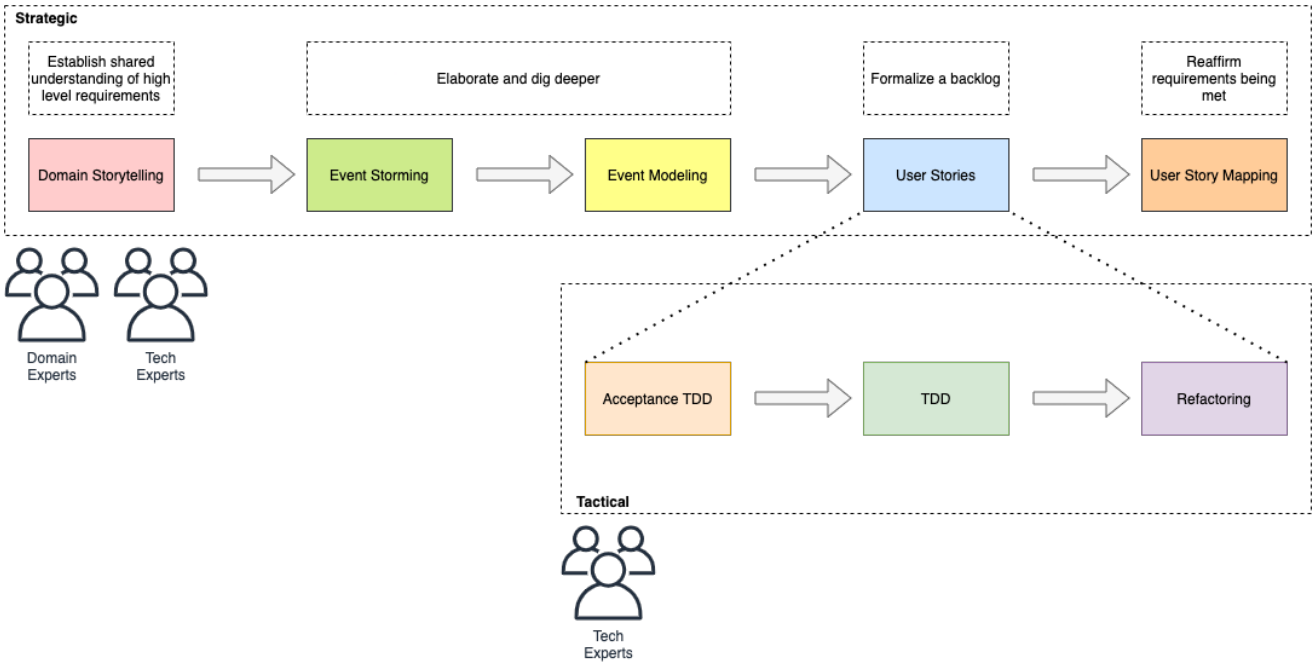

Figure 14. Domain requirements elaboration continuum

In subsequent chapters we will examine the other techniques in more detail.

Summary

In this chapter we examined two ways to enhance our collective understanding of the problem domain using two lightweight modeling techniques — domain storytelling and eventstorming.

Domain storytelling uses a simple pictorial notation to share business knowledge among domain experts and technical team members. Eventstorming, on the other hand, uses a chronological ordering of domain events that occur as part of the business process to gain that same shared understanding.

Domain storytelling can be used as an introductory technique to establish high level understanding of the problem space, while eventstorming can be used to inform detailed design decisions of the solution space.

With this knowledge, we should be able to dive deeper into the technical aspects of solution implementation. In the next chapter, we will start implementation of the business logic, model our aggregate along with commands and domain events.

Questions

1. When should you use domain storytelling?
2. Pick an application in your current context. Can you use domain storytelling to capture actors, work objects and activities for the scenario you picked?
3. When should you use eventstorming?
4. Pick an application in your current context. Can you use eventstorming to capture domain events, actors, actions, hotspots, query models, external systems, etc. for the scenario you

picked?

Further reading

Title	Author	Location
Domain Storytelling	Stefan Hofer and Henning Schwentner	https://leanpub.com/domainstorytelling
An Introduction to Domain Storytelling	Virtual Domain-Driven Design	https://www.youtube.com/watch?v=d9k9Szkdprk
Domain Storytelling Resources	Stefan Hofer	https://github.com/hofstef/awesome-domain-storytelling
Introducing EventStorming	Alberto Brandolini	https://leanpub.com/introducing_eventstorming
Introducing Event Storming	Alberto Brandolini	https://ziobrando.blogspot.com/2013/11/introducing-event-storming.html
Event storming for fun and profit	Dan Terhorst-North	https://speakerdeck.com/tastapod/event-storming-for-fun-and-profit
EventStorming	Allen Holub	https://holub.com/event-storming/

Answers

1. Refer to section Introducing Domain Storytelling
2. Share and validate the domain storytelling artifact you created with your teammates.
3. Refer to section Introducing EventStorming
4. Share and validate the eventstorming artifact you created with your teammates.