

Leveraging Event Storming for Enhanced Product Development in FinTech: A Collaborative Approach to Simplifying Complex Business Processes

Sachin Gadiyar

USA

ABSTRACT

Event Storming is a collaborative technique designed to address complex business processes in Fin Tech, involving key stakeholders such as product leaders and software engineers. This methodology simplifies intricate business requirements, rendering them comprehensible to both technical and non-technical participants. Central to this approach is the mapping of a customer's journey through four primary components: Events (what occurs), Commands (triggers for events), Systems (external applications), and Policies (rules governing outcomes).

Implementation begins with defining the boundaries of the customer journey, identifying key events, and outlining corresponding commands and policies. External systems and API interactions are also incorporated to create a comprehensive event map that aligns product and technology perspectives. While Event Storming fosters collaboration, enhances visual understanding, and accelerates prototyping, it can be time-consuming and necessitates skilled facilitation. Challenges such as scope creep and potential over complication must be effectively managed to optimize its benefits. Overall, Event Storming is a valuable tool for modeling and understanding complex Fin Tech systems, balancing its advantages against potential drawbacks for successful implementation.

*Corresponding author

Sachin Gadiyar, USA.

Received: October 06, 2024; **Accepted:** October 13, 2024, **Published:** October 20, 2024

Keywords: Event Storming, Business Processes, Product Development, Customer Journey, Events, Commands, Systems, Policies

Introduction

Event Storming is an effective technique for resolving complex business processes. In a Fin Tech context, teams typically comprise a Product Leader, software engineers, and various stakeholders. This methodology encourages collaborative brainstorming sessions aimed at achieving specific business outcomes.

Benefits for Product Development

- Accessibility:** Event Storming simplifies complex business requirements into relatable terms, facilitating understanding among both technical and non-technical stakeholders.
- Efficiency:** By centering on the customer journey, this method enables teams to derive outcomes in straightforward language. Involving end-users can further enhance understanding of user sentiment and inform decisions.

Components of Event Storming

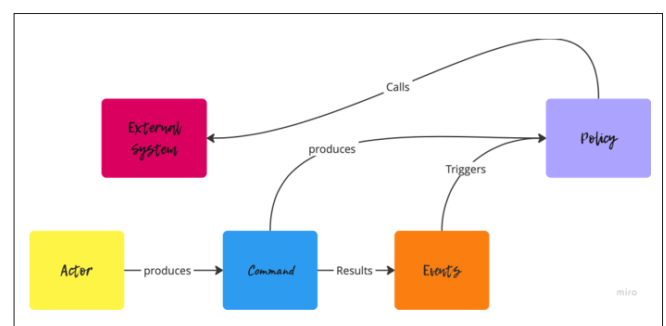
The core components of the Event Storming model include Actors, Events, Commands, Systems, and Policies:

- Actor (Yellow):** Represents the user of the system.
- Events (Orange):** Denote occurrences within the product's end-to-end journey, leading to reactions.
- Commands (Blue):** Identify the triggers for events, representing user or system actions.
- System (Pink):** Illustrates external applications and third-party services interacting with the domain.
- Policy (Purple):** Defines the rules that determine the success

or failure of the overall capability.

Together, these components effectively map the cause-and-effect processes within the system.

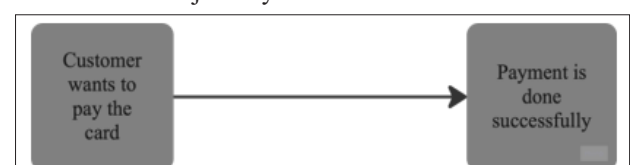
Below is a quick representation of how an event storming map can be represented.



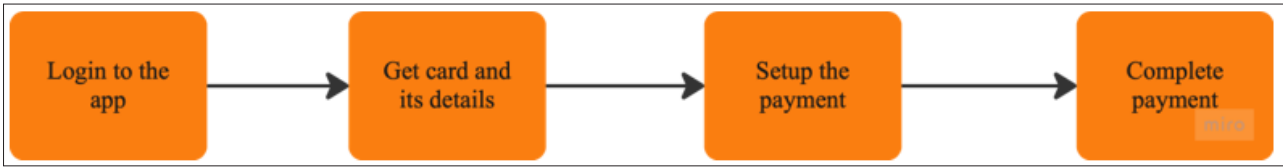
Creating an Event Map

Consider a scenario where a customer wishes to make a payment on their credit card using the issuer's mobile application:

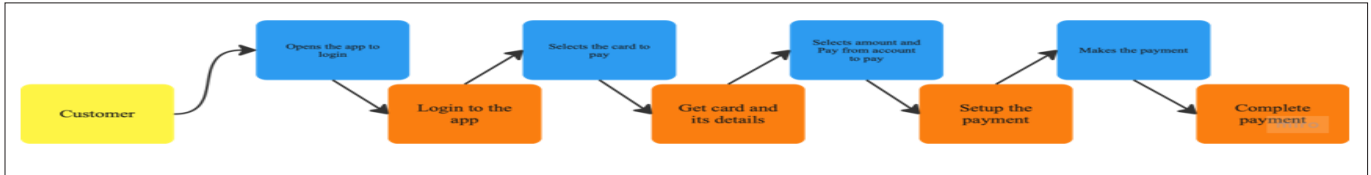
- Identify the Boundary:** Define the starting and ending points of the customer journey.



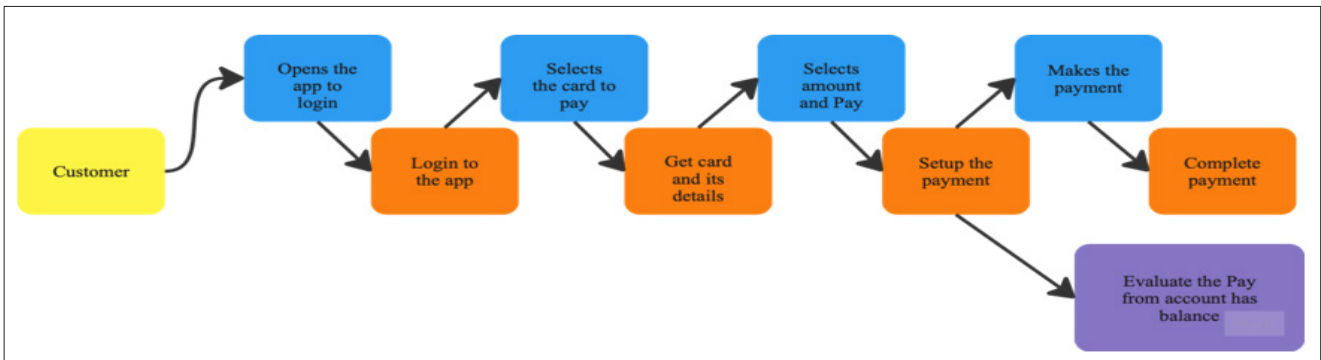
2. **List Events:** Arrange the events in chronological order. The customer journey to pay the card balance can be split into the below 4 events.



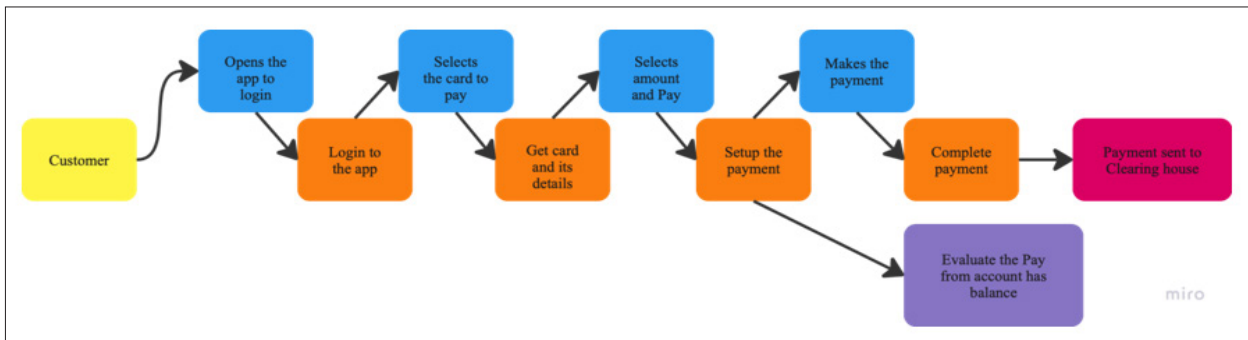
3. **Add Actors and Identify Commands:** Specify the actions taken by the customer. Commands detailed below will indicate the action the customer takes on their mobile to pay the card successfully. To fulfill each of the 4 events, customer takes 4 different actions that are listed as 4 commands below.



4. **Introduce Policies:** Establish rules necessary for successful payment processing. Policy below represents some rules the system must run to successfully complete the paying the card process. In the below case there is 1 policy detailed where the system makes sure the account the payment is happening from has enough money to pay for the card.



5. **Identify External Systems:** Recognize external entities involved in the transaction. In the below example an external system is represented to indicate that the full E-2-E processing of the payment will involve an external entity in the form of Clearing house to complete the settlement of the funds on the card and the pay from account.

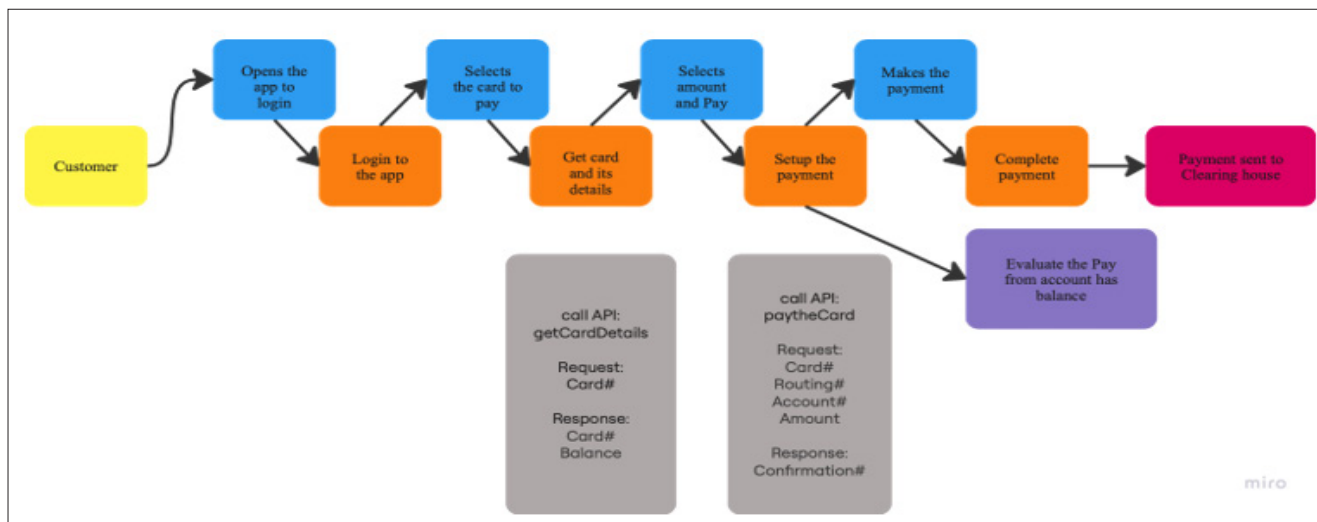


API Interactions

Integrating API interactions into the event map illustrates the interplay between customer journeys and technology. For instance, invoking an API such as get Card Details to retrieve card information exemplifies this integration.

In the example Event: Get Card and its details is brought to life by the invocation of an Api called *getCardDetails* on the mobile app. Calling the API with Request: card # provides the data elements in *Response: Card#* and *Card balance* so it can be displayed on the app UI.

Similarly, Event: Setup the Payment the Api *payTheCard* is invoked and in the Api, response is a payment *Confirmation#* indicating a successful payment done which can be displayed on the UI.



Pros and Cons of Event Storming

Pros:

1. **Collaborative Nature:** Encourages participation from diverse stakeholders, fostering a shared understanding of the domain.
2. **Visual Representation:** Provides a clear visual model of the system, making it easier to identify gaps and opportunities for improvement.
3. **Focus on Domain Events:** Centers discussions around domain events, enhancing the identification of key business processes and requirements.
4. **Rapid Prototyping:** Facilitates quick iterations on ideas without the burden of extensive documentation.
5. **Enhanced Communication:** Reduces misunderstandings through a common visual language.
6. **Encourages Critical Thinking:** Promotes thorough analysis, helping teams challenge assumptions and uncover hidden complexities.

Cons:

1. **Time-Consuming:** Initial sessions may take considerable time, especially for participants unfamiliar with the technique.
2. **Requires Facilitation:** Effective Event Storming necessitates a skilled facilitator to maintain focus, which may pose a resource challenge.
3. **Risk of Overcomplication:** Discussions can become convoluted if not managed properly, leading to complex models that may be difficult to implement.
4. **Limited Documentation:** A focus on visual models may result in insufficient written documentation for future reference.
5. **Scope Creep:** Discussions can easily drift into unrelated topics, complicating the management of scope.

Conclusion

Event Storming serves as a powerful and versatile tool for modeling complex business processes within the Fin Tech sector. By fostering collaboration among diverse stakeholders and simplifying intricate requirements into visual formats, it enhances both understanding and communication across technical and non-technical participants. This methodology allows teams to effectively map customer journeys, identify key events, and integrate external systems, leading to a more cohesive development process [1-3].

Despite its significant advantages, such as improved efficiency and rapid prototyping, the implementation of Event Storming does come with challenges, including the potential for scope creep and the necessity for skilled facilitation. Therefore, it is crucial for teams to approach Event Storming with a clear framework and defined objectives to maximize its benefits while mitigating risks. Ultimately, when executed thoughtfully, Event Storming can lead to innovative solutions and a deeper understanding of user needs, paving the way for successful product development in the dynamic landscape of Fin Tech.

References

1. Lucidchart (2024) "Event Storming 101: Accelerating Your Software Development in Domain-Driven Design" Accessed: <https://www.lucidchart.com/blog/dd-event-storming>.
2. Medium (2024) "Unleashing the Power of Event Storming." Accessed: <https://medium.com/javarevisited/unleashing-the-power-of-event-storming-e475be7826cc>.
3. Sourced Group (2024) "Event Storming." Accessed: <https://www.sourcedgroup.com/resources/tutorial/event-storming-tutorial/>.

Copyright: ©2024 Sachin Gadiyar. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.