

Product

Version

Date

User Roles

Who uses the product, and what do they get from it?

Actions

What do Actors do with the product?

Business Objects

What things does the product represent and how are they related?

External Systems

Which systems does the product exchange data with?

Reports

What do actors need to see, and when must they be told?

Non-Functional Requirements

What qualities and constraints must the product meet?

User Roles

Who uses the product, and what do they get from it?

Roles that a User (or their agent) can take on when using the product. Users can hold multiple roles beyond their basic USER role. Such Roles can represent the needs and activities of primary or supporting users. Each Role belongs to an organisation and has a number of Actions that it aims to invoke.

EXAMPLES:

User, Office Manager, Administrator, Anonymous, ...

DISCOVERY QUESTIONS:

1. Who is trying to use the system, how does their use differ?
2. What do these users need and try to achieve in a given Role?
3. Which organisation (customer, vendor, partner, regulator) does this Role belong to?
4. How many people are expected to hold this Role?
5. What are their needs, pains and gains and the value propositions?
6. What is the cost and revenue model per Role?
7. Which roles are internal, and what do they need to run the product?

Actions

What do Actors do with the product?

Actions are the interactive ways Actors engage with the system as part of the service journeys (Service Blueprints) and grouped into Use Cases and Capabilities to be delivered by functional Modules. Actions can be defined through Business Process Maps or similar sequence and orchestration diagrams defining process and screen flows, API architectures and access controls.

EXAMPLES:

Login, Reset password, Update email, Create a [object], Cancel subscription, etc.

DISCOVERY QUESTIONS:

1. What are the core Actions each Role does from start to finish of their journey?
2. What are support Actions needed for a successful journey, like undo, edit, cancel?
3. What actions does the system do on its own and are these triggered by time or event?
4. What are some potential alternatives, failure scenarios or workarounds?
5. What kind of administration actions are needed?
6. How frequently are these Actions taken, which are critical or urgent?
7. Can these Actions be monetised?
8. How would these Actions group into Use Cases, Capabilities and Modules?

Business Objects

What things does the product represent and how are they related?

Key Business Objects representing the domain of the product with their key information that Users will want to see and edit. We can show their relationships through a Domain Model. On this canvas we do not want to break them down into detail, but the technical team can translate this into detailed data schema (e.g. Entity-Relationship Models or Property Graphs) and Data Architecture Diagrams to define data lifecycle, data access, and data persistence in more detail.

EXAMPLES:

User Profile, Subscription, Subscription Payment, etc.

DISCOVERY QUESTIONS:

1. What are the things people talk about, when describing the business?
2. What are some key properties of these things that matter to the Users?
3. How do these things relate to each other?
4. Which state does each of these objects go through and what action moves it?
5. Where do these objects come from? External systems or this product?
6. How long must we store these objects? Do we need to back them up?

External Systems

Which systems does the product exchange data with?

Integrating External Systems allows us to receive or send data or utilise them to provide standard functionality that we prefer to buy instead of building from scratch. These integrations can be uni-directional or bi-directional and happen on demand via APIs or through batch data exchange.

EXAMPLES:

Authentication system, User Management system, Notification system, Payment system

DISCOVERY QUESTIONS:

1. What External Systems do we need to connect to get or send data?
2. What External Systems do we use to deliver functionality?
3. For each does data flow in or out or both ways, where is the source of truth?
4. Which organisation do these External Systems belong to?
5. How are we integrating these External Systems? APIs, batch jobs, etc.?
6. What happens if a system is unavailable?

Reports

What do actors need to see, and when must they be told?

Reports are read-only views on system activities taken based on instrumentation and logging, in order to inform or notify Actors so they can take follow-up actions as they deem fit. We distinguish between Status Reports, Audit Reports and Usage Reports.

EXAMPLES:

- Status Reports: Show live status of the system, e.g. system health, payment status, ...
- Audit Reports: Show historic actions for governance, e.g. access log, change log, ...
- Usage Reports: Show usage analytics and insights, e.g. drop-off points, app utilisation, ...

DISCOVERY QUESTIONS:

1. What kind of status does each Role need to see, what about Users and Admins?
2. What kind of governance records do we need and how long to store these?
3. What kind of product analytics do we need?
4. What kind of events do we need to track for notification?
5. What kind of events do we need to track for billing?
6. What kind of errors do we need to track?
7. Are there limits on data visibility? Is there a need to export the data and if so, what format?
8. What update frequency is needed for each Report?

Non-Functional Requirements

What qualities and constraints must the product meet?

Key Non-Functional Requirements for security, compliance, performance, scale, infrastructure, etc. (e.g. see ISO/IEC 25010 standard) in terms of orders of magnitude to keep in mind when designing the system. Ensure that the overall dimensioning and complexity is defined at the right scale.

EXAMPLES:

Response time, authentication protocols, delivery channels, localisation, uptime, data security, ...

DISCOVERY QUESTIONS:

1. Scale: How many organisations, users, objects and transactions, now and in three years?
2. Performance: Which Actions need to be instant, what can be batched?
3. Reliability: What is the impact if the system goes down? What data needs to be recovered?
4. Security: What data is sensitive and who can access it?
5. Compliance: Which laws, standards or certifications apply, where must data be stored?
6. Channels: What devices will the product support natively and in a browser?
7. Localisation: What languages and time zones are to be supported?