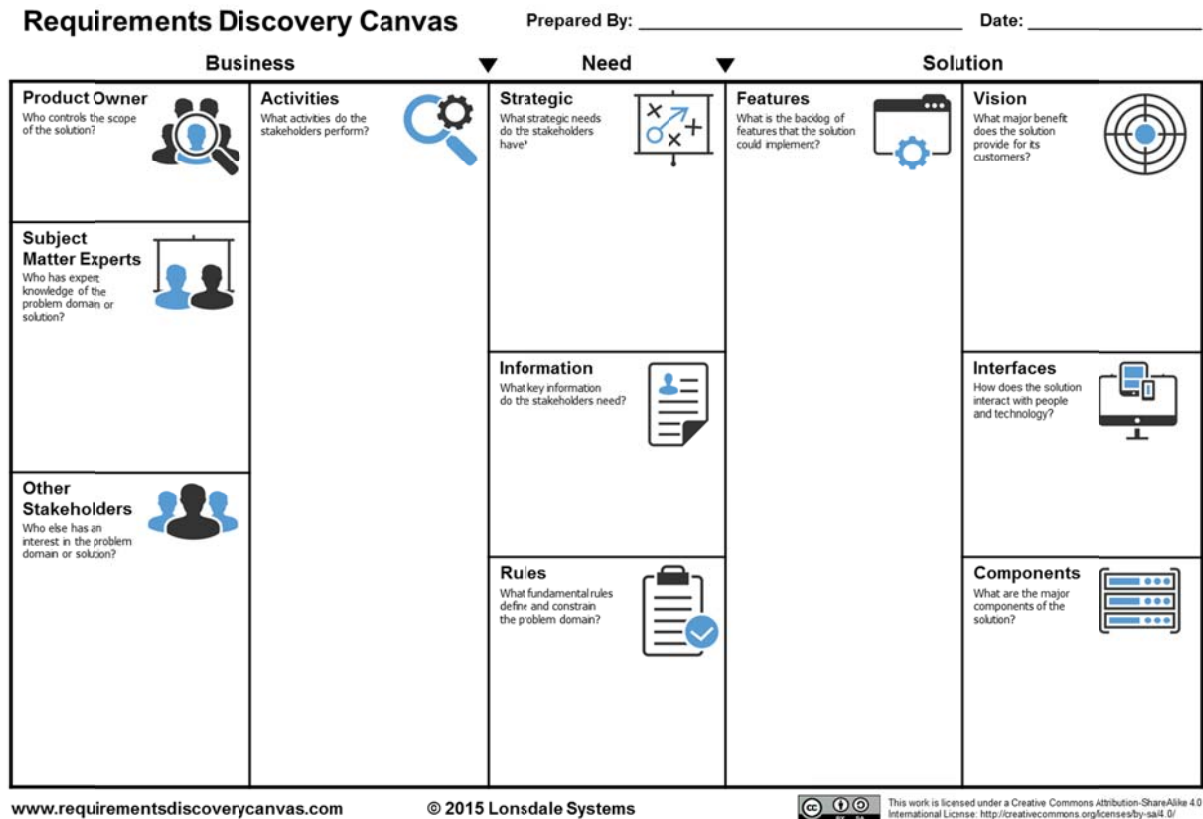


# The Requirements Discovery Canvas

The Requirements Discovery Canvas [1] is a visual tool that helps teams discover and organise software requirements. Inspired by the Business Model Canvas [2], it provides a framework for collaboration that can be used by both agile and traditional software development teams.



## Requirements Discovery Is Not Easy

Few software teams find requirements discovery a simple task. For many years, practitioners and academics alike have been searching for a universal approach to requirements that will guide a team through the discovery process. Starting with structured analysis, we have moved through successive eras of information engineering, object-oriented analysis, use cases and user stories. Teams have embraced each new era with great enthusiasm until they once again learn that requirements discovery is not a simple task!

Recently, many teams have been having greater success following an agile approach. The iterative approach to delivery is the most obvious difference between the agile and traditional approaches. But from a requirements perspective, the greatest strength of the agile approach is that a comprehensive set of requirements is no longer needed.

Instead of creating detailed requirements, agile teams capture their understanding of what stakeholders need in a list of high-level features called a product backlog. Features are elaborated into detailed requirements just before they are pulled off the product backlog and scheduled for the next iteration. The iterative approach to delivery means that there is always a working “prototype” of the solution to guide the next round of discovery.

For most teams the agile approach to requirements works well. If it has a weakness - it would be the lack of advice on how to initially populate the product backlog.

## The Requirements Discovery Canvas

The Requirements Discovery Canvas is designed to fill this gap and provide teams with a context for populating a product backlog. It helps to ensure that the product backlog doesn't degenerate into an unruly wish list of random features. The canvas also provides teams with an ongoing context for change. Traditional teams also find the canvas useful as a front-end tool for planning and scoping projects.

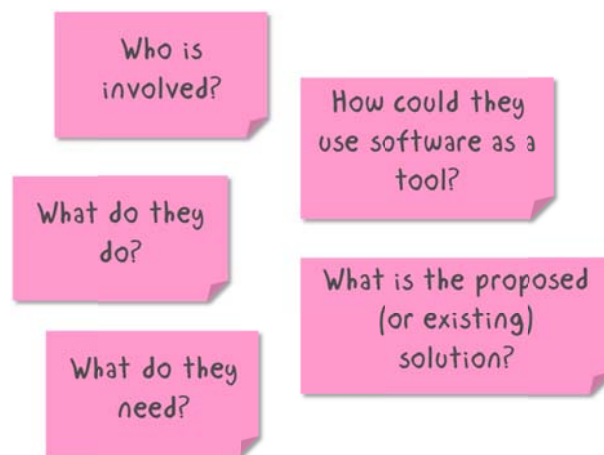
### How the Canvas Helps

The Requirements Discovery Canvas guides teams through the discovery process in two ways. First, it prompts the team to consider some fundamental requirements questions. Secondly, it serves as a visual framework for organising what the team discovers.

The canvas is not tied to a particular requirements philosophy or approach. Teams are free to choose the approach that suits them best. They can choose between a simple approach guided by the structure of the canvas. Or they can use the canvas in conjunction with a more elaborate approach if they wish.

#### Fundamental Requirements Questions

The canvas is based on a short list of fundamental requirements questions.



- Who is involved? This question prompts the team to identify the stakeholders who have some interest in a proposed (or existing) software solution.
- What do they do? This question prompts the team to identify the activities performed by stakeholders and use these as a way of understanding stakeholder goals.
- What do they need? This question prompts the team to identify the business needs that support stakeholder goals.
- How could they use software as a tool? This question prompts the team to define a set of software features that support stakeholder goals and satisfy their business needs.
- What is the proposed (or existing) solution? This question prompts the team to identify the most important components of a solution.

The five questions provide the headings for the columns of the canvas, with the body of the canvas providing space for teams to organise their discoveries. A detailed tour of the canvas follows, with examples of how it is populated. The examples are drawn from a well-known case study based on a fictitious car rental company EU-Rent [3].

## Who Is Involved?



Subject matter experts (SMEs) are stakeholders that possess expert knowledge of either a business area, or some aspect of a proposed (or existing) solution. SMEs are a valuable source of requirements but the information they provide can be influenced by politics and self-interest.

It is not unusual for business area SMEs to place their own interests above those of others when it comes to prioritising requirements. Solution SMEs have a tendency to ignore business value and focus on solution architecture when planning solution delivery. It is not surprising that there is often conflict between different groups of stakeholders.

The product owner role is responsible for resolving stakeholder conflict. Product owners ensure that a solution deliver business value to all stakeholder groups - not just the most vocal.

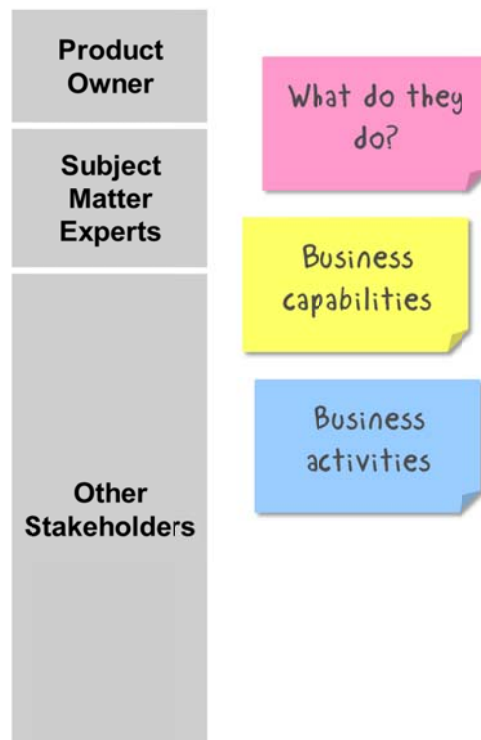
For agile teams, the product owner is usually a business representative who serves as a full-time member of the team. For traditional projects, product ownership is often provided by a steering committee. Members are selected so that there is balanced representation of all business areas.

Potential stakeholders at EU-Rent (the organisation featured in the case study) include its parent company EU-Corporation; sister companies EU-Fly and EU-Stay; the branch manager and booking clerks at each of the 1000 branches; the call centre manager and call centre operators; and the manager and mechanics working at the maintenance centres.

It is not difficult to identify potential SMEs among the stakeholders at EU-Rent. Choosing a suitable product owner however would not be so easy. Balancing the needs of the branches, the call centre and maintenance centres, not to mention those of the parent and sister companies, is likely to present a challenge.

As well as the product owner and SME's, there are other stakeholders that must be considered. Regulatory bodies exert a strong influence on a solution when there is a need for compliance. As well as SMEs, there will be less expert stakeholders who will have requirements to contribute. Stakeholders affected by a proposed solution will need to be identified and informed of the expected impact.

## What Do They Do?

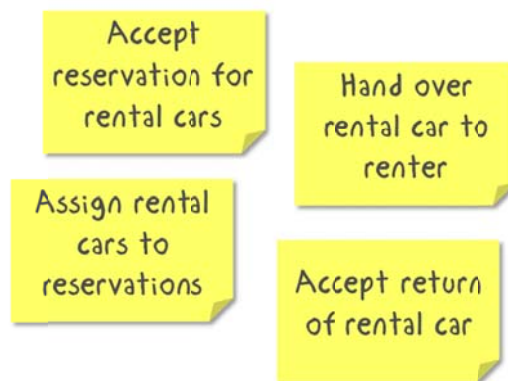


It is widely accepted that a team should understand stakeholder goals, before they attempt to define a software solution to meet their needs. However, understanding stakeholder goals is not always easy. Stakeholders often respond to direct questions with high level, generic goals.

For example asking the question - "What are the goals of EU-Rent?", is likely to produce responses such as: to be profitable; or to ensure customer satisfaction. Neither of which will increase the team's understanding of the business area.

Based on experience, it is often better to follow an indirect approach to understanding stakeholder goals. An approach that works well is to question the stakeholders about the activities they perform.

Asking, "What activities are performed at EU-Rent?", is likely, with a little prompting, to produce a much clearer response.



As these examples illustrate, activities are best named using an active verb and noun phrases. The names should focus on "what" the stakeholders do rather than "how" they do it. Following these simple guidelines also makes it easier to check the expected outcome of an activity.

## The Requirements Discovery Canvas

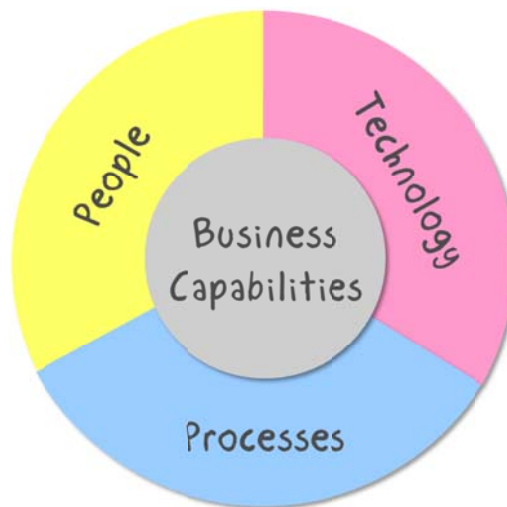
For example the EU-Rent activities can be reworded as outcomes.



Stakeholders should find it easy to confirm that these outcomes are important and that they represent the goals of their business area.

When a business area has a large number of activities, it is convenient to group them into business capabilities. These describe a business area's ability to repeatedly perform a group of activities.

At first sight, business capabilities might seem rather similar to business processes, but they are not. Business capabilities describe the core capabilities of a business area, in contrast, business processes describe the realisation of capabilities as workflows involving people and technology.

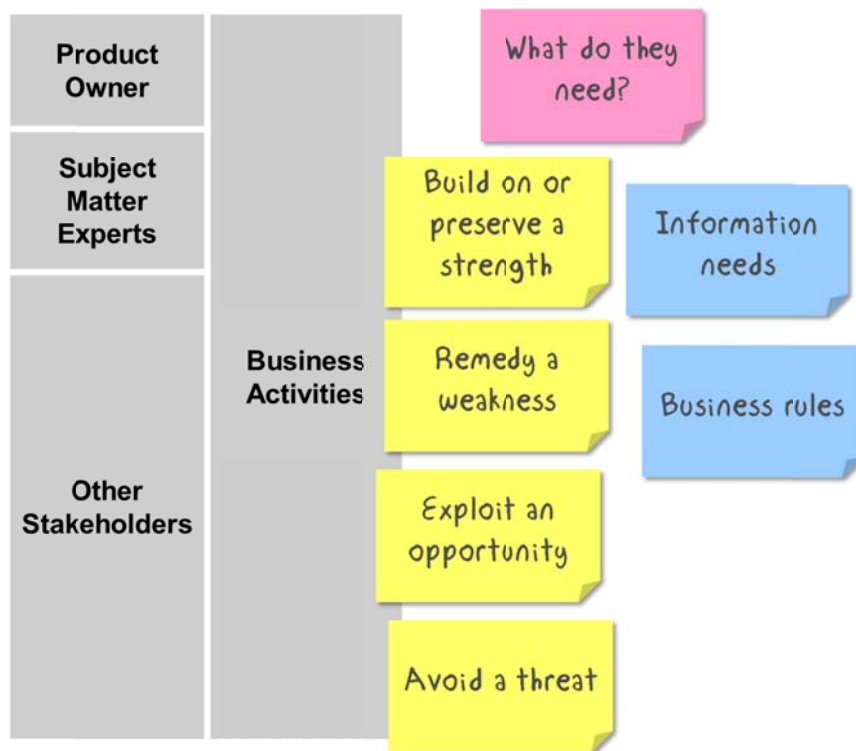


## The Requirements Discovery Canvas

Rather than the verb and noun phrases used to name activities, capabilities are best named using noun phrases. For example, the activities performed at EU-Rent could be grouped into the following capabilities.



## What Do They Need?



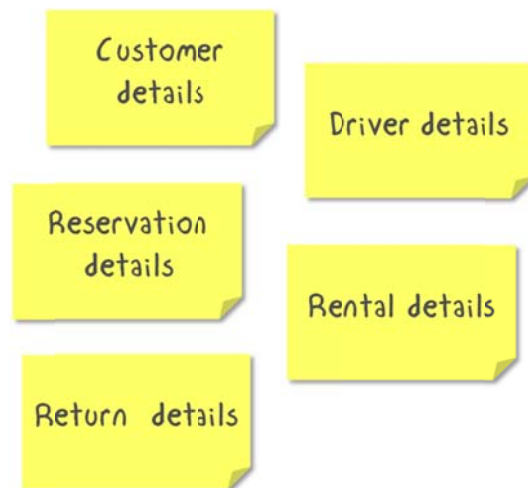
Business need is rather a vague term that suggests many different types of "need". All business areas have a need for staff, finance, equipment etc. In the context of the Requirements Discovery Canvas, a business need is something that can be satisfied by a software feature.

The two most common business needs satisfied by software are:

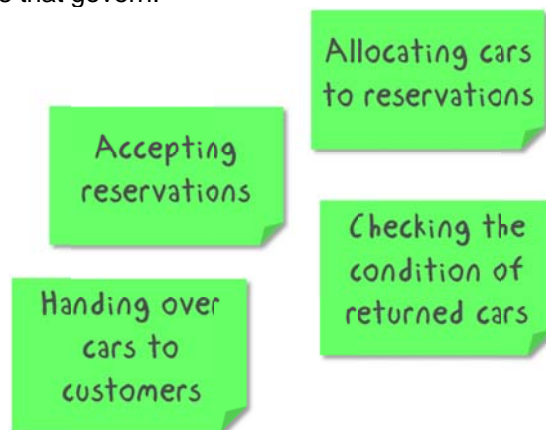
- the need to store, retrieve and manipulate information; and
- the need to enforce business rules that define, constrain or guide some aspect of a business.

## The Requirements Discovery Canvas

EU-Rent needs to store information about:



And enforce business rules that govern:



Managing information and enforcing business rules are operational business needs that often provide value to an organisation by increasing its revenue, avoiding costs or ensuring compliance.

As well as satisfying operational needs, software solutions can be designed to meet the strategic needs of a business area. Strategic needs often emerge from a SWOT analysis that identifies the need to:

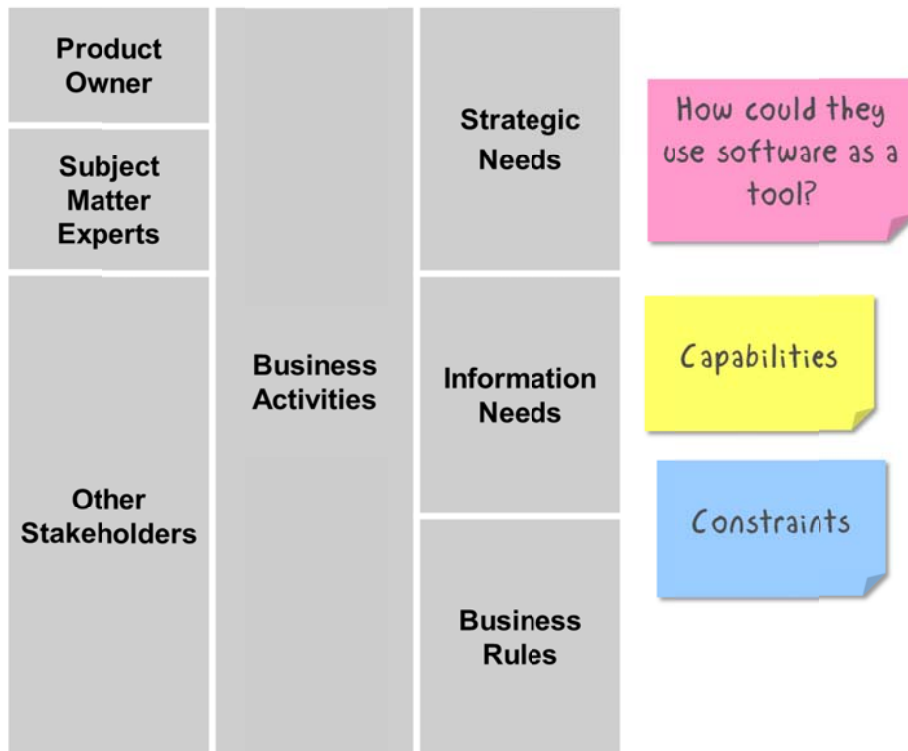
- build on or preserve a strength;
- remedy a weakness;
- exploit an opportunity; or
- avoid a threat.

For example, EU-Rent might discover that some branches are turning away customers because they don't have cars available. While other branches have a large number of surplus cars resulting from "no-shows" (customers who make reservations but never turn up to collect the car).

EU-Rent might seek to remedy this weakness in a number of different ways. They could build an application to track the transfer of cars between branches; they could develop an algorithm to predict the number of no-shows; or they could develop a mobile application that allowed potential "walk-ins" (customers without a reservation) to check the availability of rental cars at each branch.

Satisfying strategic needs often provides far greater value to a business area compared to meeting operational needs. The strategic use of software can provide value by strengthening a business area's position in a market or by improving customer service.

## How Could They Use Software As a Tool?



Software features are short statements describing a software capability or constraint. Stakeholders should be able to recognise the purpose of a feature and confirm that it satisfies one or more of their needs.

Capabilities describe the ability to:

- store and retrieve data;
- interact with users;
- interact with external systems and devices; and
- enforce business rules.

Constraints are restrictions that can be placed on individual capabilities; or limit the entire solution in some way. Constraints often describe qualities such as reliability, usability, security or performance.

Some of the features satisfying business needs at EU-might include.





## The Requirements Discovery Canvas

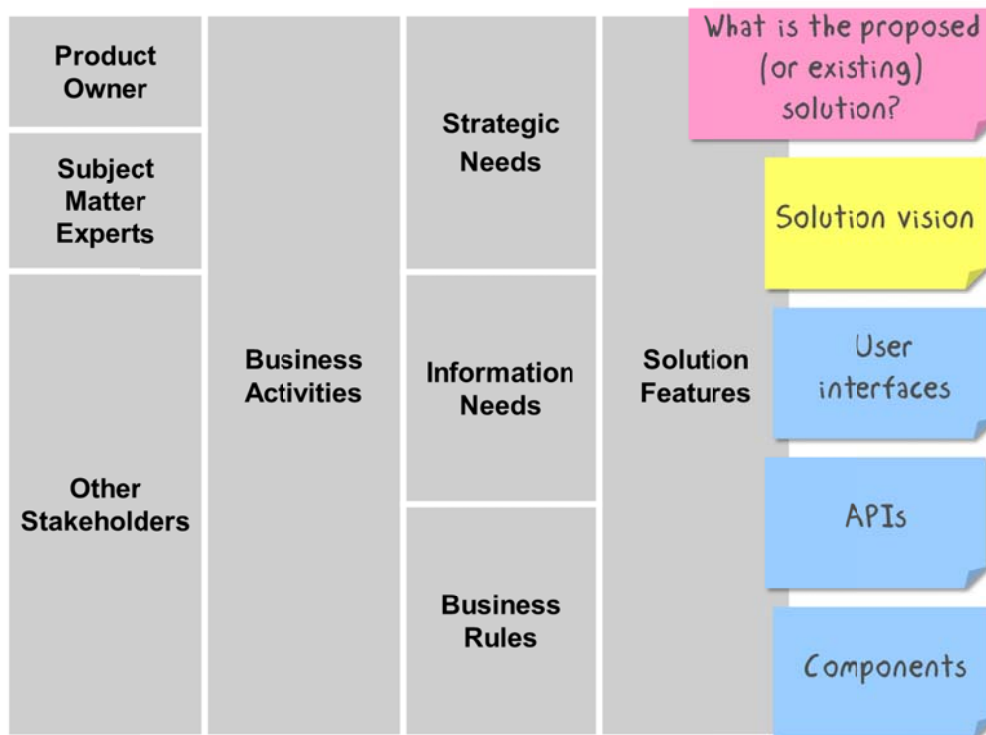
Compared to detailed software requirements, lists of features are much easier to work with. It is a simple matter to sort and filter a list to focus on and explore different aspects of a proposed solution. Because they understand their purpose, stakeholders are able to prioritise a list of features to reflect the business value offered by a feature.

For agile teams, a prioritised list of features provides a product backlog, which they use as a basis for estimating, and planning delivery. Agile teams often elaborate features into user stories, acceptance criteria or scenarios that can be implemented as automated tests using tools such as Cucumber.

For traditional teams, features can provide early estimates of a project's scope. Traditional teams often elaborate features into use cases, scenarios or natural language statements.

Both types of team can use a list of features to prepare long term plans such as product road maps and release plans.

### What is The Proposed (Or Existing) Solution?



All teams benefit from developing a concise, high-level vision for a solution they plan to deliver. The vision is often described using a short structured statement that identifies:

- the target customer for the solution;
- the primary need that will be satisfied;
- the key benefits provided by the solution; and
- how the proposed solution differs from other approaches.

Support teams understand the benefit of identifying the components, APIs and user interfaces of an existing software solution. They know that this understanding will provide them with a baseline on which to develop future enhancements.

## The Requirements Discovery Canvas

Teams developing new solutions may find it harder to understand how the details of a proposed solution could influence requirements. This is especially true for teams that are used to defining solution-independent requirements. But there are several aspects of modern software solutions that can influence requirements.

New technologies, such as the cloud and mobile devices, make possible software features that would have been impractical in the past. For example, combining cloud based APIs with mobile devices now makes it practical to synchronise a central database across many devices.

Many software applications are now based on a mix of technology such as client-server, browser-based and batch applications. These applications are deployed on a range of platforms that can include desktop PCs, in-house servers, cloud based servers and mobile devices. Within this mix, the choice of technology can have a significant impact on the feasibility of a required feature.

In some cases, it might be required to implement the same feature on a number of different platforms. The unique capabilities and constraints of each platform might require that slightly different version of the feature are implemented on each platform. In other cases, the implementation of a feature may be distributed across several platforms. This creates new requirements for APIs to interconnect the various platforms.

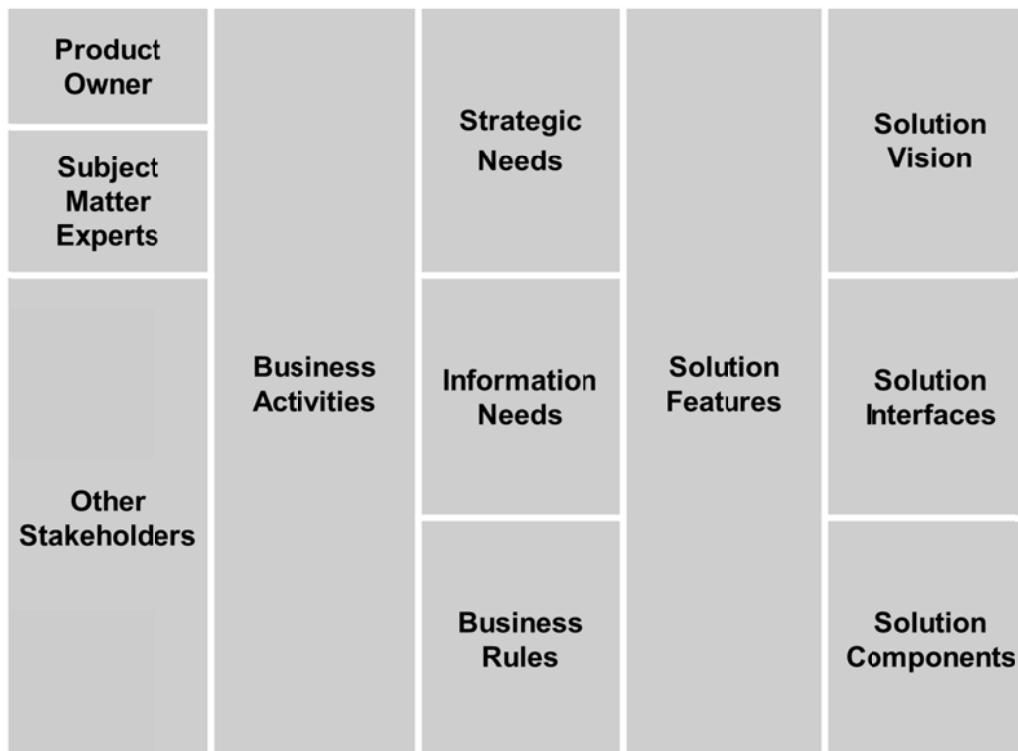
The popularity of packaged solutions is another reason why teams might need to focus on the details of a solution. Software packages must be configured to satisfy stakeholder needs, but before this can take place, teams must identify and understand the major components of the package and the features they provide.

Finally, the widespread use of software by people with varying levels of skill, has placed greater emphasis on interaction design. Many customers now experience all or part of an organisation's services through a user interface. Poor design of user interactions can reflect badly on services and adversely affect buying decisions. This means that many organisations now view interaction design as an essential starting point for requirements.

These aspects of modern software solutions means that a purely solution independent view of requirements is less practical than in the past.

## The Complete Canvas

We have come to the end of the tour with the complete Requirements Discovery Canvas presented below. A blank template [4] of the canvas is available for download.



The Requirements Discovery Canvas offers a visual framework for identifying and organising requirements. The canvas facilitates face-to-face and remote collaboration, which promotes a shared understanding of requirements.

Agile teams can use the canvas as a tool for populating and managing a product backlog. Prioritised features can later be transferred from the canvas to Sprint Task or Kanban boards to plan and manage delivery of the solution. As each increment of the solution is delivered, the canvas can be updated, providing traceability to requirements already delivered and a context for the next solution increment.

Traditional teams can use the canvas, to plan the scope of a project and outline the contents of more detailed requirements documents. Documents can be organised to reflect the structure of the canvas so that they build on the shared understanding promoted by the canvas.

Another possibility is to use the canvas to organise and index a series of smaller, more focused requirements documents, thus avoiding many of the problems associated with large, unwieldy requirements specifications.

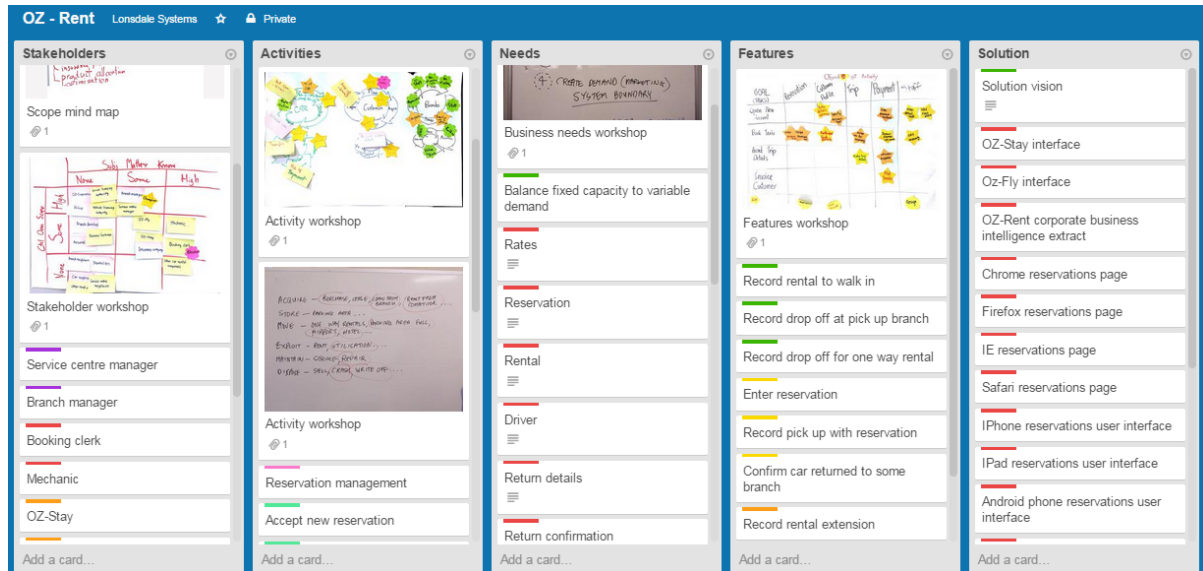
## Workshops

As the case study examples have helped to show, populating the canvas with coloured sticky notes is an ideal activity for requirements workshops. If wall space permits, the canvas might also become a permanent artefact providing a visual requirements dashboard visible to the entire team.



## Tools

The simple structure of the canvas means that teams who collaborate remotely, can take advantage of tools such as Trello that mimic working with sticky notes. Trello's ability to attach images to cards makes it a great tool for organising photos of white boards and flip charts taken during requirements workshops. The ability to also attach documents means that Trello can be used as a visual index into more detailed requirements documents and models.



An example of how Trello can be used in conjunction with the canvas can be found at [trello.com/b/IFtlbcV5/oz-rent](https://trello.com/b/IFtlbcV5/oz-rent). The example also provides more detailed examples based on the EU-Rent case study.

1. [www.slideshare.net/lonsdalesystems/requirements-discovery-canvas](http://www.slideshare.net/lonsdalesystems/requirements-discovery-canvas)
2. [www.businessmodelgeneration.com/canvas/bmc](http://www.businessmodelgeneration.com/canvas/bmc)
3. [www.businessrulesgroup.org/first\\_paper/br01ad.htm](http://www.businessrulesgroup.org/first_paper/br01ad.htm)
4. [www.slideshare.net/lonsdalesystems/requirements-discovery-canvas-a3](http://www.slideshare.net/lonsdalesystems/requirements-discovery-canvas-a3)