

[Click Here](#)



Use cases examples

When projects involve multiple stakeholders, it's important to communicate information efficiently and effectively. Whether it's describing how a product works or how customers interact with it, clearly representing information can make all the difference. A use case is one way to share information. They simplify complex ideas and offer a way to share information visually. While software development often works with use cases, businesses across industries can benefit from them. Let's learn more about the topic, including use case examples. A use case is a methodology to organize system requirements. It comprises a set of possible sequences of interactions between systems and users in a defined environment with a particular goal. The use case document includes the environment, goal, and other details to provide a clear picture of the system and overall project. Use cases are invaluable for software developers, helping them identify potential errors in the system and fix them early on. They're also incredibly beneficial for project management, offering context to anyone involved, from stakeholders to customer service representatives. Use cases vs. user stories Use cases and user stories can benefit project management and development teams. However, distinct differences exist between the two. User stories are short-form descriptions from a customer's perspective. They are the story's beginning, setting the stage for a specific feature or workflow. Use cases offer greater context, including: The goal of the use case The series of paths the system could take Post-conditions, defining the actions the system takes after the use case Use case vs. test case People commonly interchange use and test cases, but they have distinct objectives. A test case is a group of test inputs and anticipated results that lead to the further development of a particular test objective. You can reuse test cases. They keep teams aligned, validate software features, and can lead to improved software quality. On the other hand, use cases focus on end-user interaction with the system, seeing how well specific users interact with a system rather than testing out software features. They are based on system requirements and support different paths, while test cases support a single outcome. Just upload your customer research and ask your insights hub - like magic. Try magic search! What's the purpose of a use case? A use case has multiple purposes. Use cases can establish requirements, outlining how users interact with the system. They can also manage scope, reducing dreaded scope creep and providing context for all development team members. One of the most obvious benefits of a use case is that it can reveal potential problems early in the process. Establishing use cases in your development project ensures you'll create a product that is better equipped to serve everyone who interacts with it. Who creates use cases? While use cases can benefit every member of a product development team, project or product managers are usually the people responsible for outlining the initial use cases. Product managers can keep the team aligned as they address and complete objectives, all while keeping the concerns of the system's users in mind. While managers typically control the document, team members can add context or additional details to the use case. Product developers often add technical and design elements to the use case, which gives the development team necessary insights into the design and creation of the product. There are two primary types of use cases: Business use cases Business use cases are also referred to as "abstract-level use cases." They tend to be high-level, with a more abstract description that refers only to the business process and actors involved. Business use cases define the actions the business must perform to provide observable results. System use cases System use cases have more detail than business use cases. They include the specific system processes that must happen to fulfill the project's goal. System use cases are also known as "implementation use cases." Companies can use both cases. Many project managers start with business use cases to provide high-level overviews before moving on to system use cases for a more granular level of detail. Six elements of a use case A typical use case contains the following six elements, whether high-level or micro-detailed: 1. Actors In the context of a use case, an actor is anyone who performs a behavior or interacts with your system. An actor doesn't have to be an individual. It can also be a company or a team. 2. Systems The system is the product that provides the backdrop for your actors' interactions. It is also called the scene or the subject. 3. Goals Goals are the results of the participating actor's interactions with the system. You can discuss these goals at the project's outset with all stakeholders and members of the development team. 4. Preconditions In any use case, it's crucial to meet certain conditions before and after the use case. These are called preconditions. They're invaluable for software developers, allowing them to visualize the steps users experience as they interact with the system. 5. Basic flow The basic flow is when a use case operates as outlined and intended, with no deviations or errors. Deviations are normal, so a basic flow is often the starting point for development. 6. Alternative flows Alternative flows are when any deviations occur from the basic flow. This can manifest when a system-level error occurs or when users interact with the system in unexpected ways. What to know about use case diagrams A use case diagram visually represents your system's actors and their interactions with the system. Use case diagrams can be particularly helpful for product developers and designers, clearly depicting actors, scenarios, and goals. To write a use case, clearly define the target audience you want to focus on. From your target audience, isolate a single user. They can serve as the actor for your use case. Work with your development team to create an outline that includes the typical flow of events when the actor uses the product or service. This is the basic flow you and your team operate from. Include alternative flows in the use case. Provide examples of how to proceed in the event of all flows so your development team has more to work with. Once the use case is complete, share it with those involved in the project. Consider the feedback you receive and incorporate it into the use case as necessary. There's no single format for use cases. Depending on the nature of your project and your company's needs, you could include or eliminate various sections. You can use this basic section outline as a starting point for your use case: Use case name Summary System details Actors/primary actor Preconditions Basic flow Alternative flows Goal Stakeholders You might also include a priority in your use case to define the project's urgency. Including a use case diagram with your outline can also be helpful to provide visual cues to any non-technical people reviewing the use case. An actor in a use case is someone who interacts with the system. To identify an actor for your use case, consider what the system handles and what is handled outside the system. Additionally, consider what the actor needs from the system and the expected outcome. A secondary actor in a use case provides a service to the system you're refining. Secondary actors do not trigger interactions with the use case and are often termed "supporting actors" by those involved in the project. A Delivery Challan is a formal document that is created in situations where goods are being transported from one place to another which may or may not result in sales. Example: Transfer of goods from the Head Office to its branches. Delivery Challans are sent along with the shipment of goods. It contains the details of items and quantity of goods being delivered. Use-Cases: Supply of goods on approval. The quantity of goods to be delivered cannot be determined during dispatch. Example: supply of liquid gas. Transportation of goods for job work. Transport of goods in a semi-assembled state or completely in parts. Guide map Enabling Delivery Challan Before creating a Delivery Challan, please make sure you have enabled the feature under Settings. To enable Delivery Challan: Click the Gear Icon at the top right corner. Navigate to the Preferences menu. Enable Delivery Challan and other required modules under the General tab. Hit Save. Creating a Delivery Challan To create a Delivery Challan in Zoho Inventory, Navigate to the sidebar. Click the '+' button adjacent to Delivery Challans. Enter the details of the Delivery Challan in the creation form. Click Save. Fields Description Customer Name The name of the recipient to whom you wish to raise a Delivery Challan. Place Of Supply The registered place of business of the selected recipient. Delivery Challan# A unique ID for every new Delivery Challan created. You can choose to either auto-generate or manually add the Delivery Challan number by clicking the Settings icon that is next to it. Delivery Challan Date Date on which the Delivery Challan is generated. Challan Type The purpose for which the Delivery Challan is being created. Statuses of a Delivery Challan Status Description Draft Delivery Challans when created will be in the "Draft" status before being marked as "Open". Open The goods have been dispatched for delivery. Delivered The goods have been delivered to the recipient. Returned The goods that were dispatched have been returned. Learn more about Delivery Challans here. Records and Return In some situations, all the goods that you transport may not go into sale. In this case, you can record the return of the goods that have been delivered but not yet billed. For instance, John has sent three boxes of raw materials to a manufacturing firm and issued a delivery challan for its transportation. After delivering, it turns out that they only require two boxes at the moment. Now, he can issue an invoice only for the boxes that his customer accepts and record a return of the other items. Here, you can learn about : Recording a Partial Return To Undo the Partial Return Recording a Partial Return After delivering the items, if your customer accepts and purchases only some of them, then you can record a return for the remaining items. Prerequisite: Some items in your delivery challan must be invoiced to be able to record partial returns. To record the return of goods in a delivery challan: Go to the Delivery Challan module on the sidebar. Open the delivery challan. Click More > Record Partial Returns. Configure the items and the quantity of those items that are returned. Click Proceed. Note: You can record only one return on a delivery challan as of now. To record another return, you must undo the first return and then re-record it. To Undo the Partial Return If you want to undo the returns recorded: Go to the Delivery Challan module on the sidebar. Open the delivery challan. Click More > Undo Partial Returns. You can now invoice those items or re-record the return. Share — copy and redistribute the material in any medium or format for any purpose, even commercially. Adapt — remix, transform, and build upon the material for any purpose, even commercially. The licensor cannot revoke these freedoms as long as you follow the license terms. Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits. You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation. No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material. In Business Analysis, understanding and effectively implementing Use Cases can be the key to driving successful projects and achieving organisational goals. Use Cases serve as practical tools that help business analysts, developers, and stakeholders identify needs, set goals, and improve collaboration. They provide a structured approach to defining the interactions between systems and users, ensuring that software and systems deliver the intended functionality. Discover how Use Cases in Business Analysis can drive successful projects across different industries. Empower your business with actionable insights and explore the power of Use Cases in Business Analysis today. Plus, learn more about their significance and application. Table of Contents 1) Use Cases and scenarios- an overview 2) Use-Case definition 3) Use-Case benefits 4) Factors influencing Use Cases 5) The anatomy of a Use-Case 6) Guidance for Use-Case template 7) Conclusion Use Cases and scenarios- an overview In Use Cases and scenarios within the Business Analysis domain, it is used to define the particular ways a system will be able to interact with the users or other systems. A Use Case is a structured explanation of the actions of the system in collaboration with external entities (users, systems, hardware), which result in providing valuable outputs. On the other hand, scenarios are specific to instances of a Use Case, detailing the steps in the sequence and user-system interaction to perform an action or goal. Understanding these concepts is crucial for business analysts, developers, and stakeholders to ensure that software and systems meet the intended requirements and provide the necessary functionality. They serve not only as guidelines for system design but also as vital tools for testing and validating the system post-implementation. Definition of Use-Case A Use Case is a fundamental element in software and system design that outlines how a system interacts with external entities to achieve a specific goal. It provides a detailed, structured explanation of the system's actions in collaboration with users, other systems, or hardware, resulting in valuable outcomes. Use Cases are essential for ensuring that systems meet intended requirements and function as expected. Title The title of a Use Case should be action-oriented and reflect the main goal or outcome. It acts as a concise summary of what the Use Case is about, such as "Process Online Order" or "Update User Profile." Actors Actors are the external entities that interact with the system. They can be primary actors, who initiate the Use Case to achieve a goal, or secondary actors, who assist or interact with the system during the Use Case execution. Examples of actors include a user, a customer, another system, or a database. Description The description provides a brief overview of the Use Case, detailing the intent and the significance of the Use Case in the broader system context. It should answer what the Use Case does and why it is important. Preconditions Preconditions specify what must be true or satisfied before the Use Case can be initiated. These conditions ensure that the system is in the correct state to perform the Use Case actions. For instance, a precondition for the "process online order" Use Case could be that the shopping cart is not empty. Main flow The Main flow is the sequence of steps that describes the typical execution of the Use Case, detailing how the actors will interact with the system to achieve the goal. This flow is often depicted in a step-by-step manner, highlighting interactions and decision points. Gain skills that can be implemented in real workplace situations- register for our BCS Foundation Certificate in Business Analysis Course today! Alternate flows Alternate flows cover the variations of the main flow. These include exceptions or error conditions that could alter the course of the main flow, detailing how the system should respond in different scenarios. For instance, an alternate flow might handle situations where an item in the order is out of stock. Postconditions Postconditions describe the state of the system at the conclusion of the Use Case execution. They define what has changed or what must be true after the completion of the Use Case, ensuring that the system's integrity is maintained. For example, the postcondition for a successful "update user profile" might be that the user's details have been updated in the database. Benefits of Use-Case Efficiency Efficiency is something indispensable in the current rapidly that marks business life, and Business Analysis will contribute substantially to the optimisation. They find bottlenecks and redundancies to propose solutions that bring improvements in the workflow by understanding processes and requirements. In most Project Management areas, such as workflow analysis and requirements gathering, the application of Business Analysis techniques can reduce the project cycle time and increase productivity. Real-world examples would be the use of automating data entry-type tasks or fine-tuning the supply chain logistics, whereby both demonstrate a significantly higher impact on wasted resources and time. Clarity Another great benefit of Business Analysis is that it clearly defines the operations within the business and the objectives of the project. The analysts ensure the objectives of the project are defined and articulated, and the key subject—who is to accomplish what in relation to project accomplishment—has clarity. All these enable the achievement of this clarity because of meticulous requirement documentation and regular communication, leading to data-oriented decisions and not subjective decisions. For example, SWOT analysis can precisely determine the strategic positioning of a project, aligning the project to broader business objectives. Check out our BCS Foundation Certificate in Organisational Behaviour today! User-centric approach Effective Business Analysis has a user-centric approach. This helps the business, in this case, to understand or focus on what the end-user needs and experiences in such a manner that they can tailor their product or service offering in the manner most beneficial to the market. This way, business analysts will gather users' feedback and focus it on refining product features and service delivery methods. The focus in this regard ensures that the final offerings are based on theoretical data and are, therefore, practical and implementable in real-life situations. Implementation of user-centric strategies largely gives firms an edge in enjoying even higher levels of customer satisfaction and loyalty, hence translating directly to more sustained growth in business. Engagement Unquestionably, there are increased stakeholders in the project, with effective Business Analysis ensuring that there is maximised engagement of the stakeholders. Business Analysts involve stakeholders in decision-making from the early development of a project right through its execution in order to maintain alignment and buy-in. Some of the techniques used include regular meetings with stakeholders and updates to make sure everybody is on board and involved in the program/project. This is supposed to be done through anticipation of probable issues and addressing them well in advance, but it also ensures that the stakeholders are nearer to their expectations, hence enhancing the chances of success of the project. Factors influencing Use Cases Before diving into the anatomy of a Use Case, it is important to understand the factors that shape its creation. Each of these factors ensures that the Use Case is well-defined, actionable, and aligned with business objectives. Stakeholders Identifying the stakeholders is one of the first critical steps in writing a Use Case. The term "stakeholders" describes persons who would be interested in the project's outcome and can refer to internal employees or managers and even external customers or partners. This roadmap to the scope and focus of the Use Case is best known by understanding what stakeholders are involved, together with their interests and requirements. Preconditions Preconditions describe the state of the system before the Use Case commences. They prepare the ground for action in such a way that all necessary conditions be fulfilled, which are indispensable for the smooth going-on of the process. Take for instance a banking application, the precondition to the "Withdraw Cash" Use Case is that the user should be authenticated and have a positive account balance. Triggers are events that set the Use Case into motion. These might be actions the user directly takes, such as clicking a button, or it might even be of an externally driven nature, such as the coming of some date or time. It helps in mapping out the user journey, understanding the triggers, and identifying how the trigger dynamic interacts with the system. Post-conditions They are used to define the state that the system gets in after the completion of the Use Case. This is to ensure that the system's response to the user action is properly defined and that the end state fulfils the business objectives. For instance, in the "withdraw cash" Use Case, after the condition, the money is to be withdrawn from the user balance, and the transaction record is to be made. Gain in-depth knowledge of various Project Management skills with our BCS Foundation Certificate in IS Project Management Courses. The anatomy of a Use-Case A well-drafted Use Case is like a blueprint for system interactions. It typically includes several key components: a) Title and identifier: A concise title that describes the action or goal, accompanied by a unique identifier. b) Primary actor: The main entity performing the Use Case. c) Goal in context: A clear statement of the user's goal and the context in which it is performed. d) Preconditions and post-conditions: As discussed, these define the initial and final states of the system related to the Use Case. e) Main success scenario: A step-by-step description of the interactions for achieving the goal under normal conditions. f) Extensions/alternate flows: Describes alternative actions and flows that deviate from the main scenario due to various conditions. Guidelines for Use-Case template Creating a standardised Use-Case template ensures consistency, clarity, and completeness in documenting system interactions. Below are guidelines for developing a comprehensive Use-Case template: 1) Use-Case identification Each Use-Case should have a unique identifier, which helps track and reference throughout the project lifecycle. This identifier might include a prefix based on the project or system, followed by a sequential number (e.g., UC001). 2) Use-Case history Documenting the history of the Use Case, including its versions, authors, and revision dates, is crucial for maintaining the integrity and relevance of the Use Case over time. This section aids in understanding how the Use Case has evolved and provides insights into the rationale behind certain changes or decisions. 3) Triggers Identifying the events that trigger the start of the Use Case, such as user actions or external events (e.g., user clicks "Submit Order" button, scheduled report generation) is crucial for understanding how and when a Use Case begins. 4) Assumptions Documenting any assumptions involving preconditions made during the development of the Use Case (e.g., users have internet access, payment gateway is operational). Conclusion Understanding Use Cases in Business Analysis is crucial for effective Business Analysis, as it provides a structured approach to defining system interactions and achieving business objectives. By following a well-defined Use-Case model and considering factors like stakeholders and preconditions, businesses can streamline processes, enhance clarity, prioritise user needs, and maximise stakeholder engagement. Inspire and motivate your team and drive success with our Creative Leaders Training. Join today! A use case is a concept used in software development, product design, and other fields to describe how a system can be used to achieve specific goals or tasks. It outlines the interactions between users or actors and the system to achieve a specific outcome. In this article, we'll dive into the details of what use cases are, how they are used in software development, and their benefits. We'll also explore common types of use cases and provide some tips on how to create effective use cases. Moreover, to help you effectively manage your projects use cases, we'll offer a pre-built requirements management template that can help you gather all the necessary information and ensure all stakeholders are aligned on the project's goals. Use cases are incredibly valuable for documenting how users interact with a product, system, or process. They capture the context around how and why your customer or user engages with a solution. Developing strong use case examples provides immense benefits throughout your design, development and testing processes. In this comprehensive guide, we'll explain what use cases are, why they matter, and provide 15 diverse examples you can reference to create effective use cases for your own products or services. Let's dive in! What is a Use Case? A use case is a methodology for documenting the interactions between an external actor (user, external system, etc.) and a process or product. Use cases capture the goals users want to achieve by using the product and how the system responds. Each use case focuses on a specific way the user engages with the system to complete a discrete task. It outlines who is performing the use case (actor) What they want to accomplish (goal) Step by step actions to achieve the goal Documenting use cases helps you understand functional requirements, design intuitive products, and ensure you build solutions tailored to your users' needs. Why Use Cases Matter Investing time in developing detailed use cases provides immense value across product design and development, including: Understanding user goals - Get inside the mind of users and what they want to achieve. Defining requirements - Capture essential features and capabilities needed. Driving design - Create optimal user flows and interfaces. Developing test cases - Derive integration and system test cases. Supporting launch - Train support teams on common user interactions. Simplifying documentation - Explain how a system works conversationally to users. Well-constructed use cases make products intuitive and user-friendly. They ensure you build the right solutions for how your customers actually work and interact. 15 Example Use Cases Now let's explore 15 use case examples covering diverse products, services, and processes. Review these samples to get ideas for developing compelling use cases tailored to your own offering: 1. Order Food Delivery Actor: Customer Goal: Order meals for delivery from a mobile app Steps: Customer opens food delivery app Customer selects location to have food delivered Customer searches for nearby restaurants Customer selects restaurant and reviews menu Customer adds desired items to order cart Customer selects checkout and enters delivery address Customer selects payment method and submits order App displays estimated delivery time and order status 2. Withdraw Cash from an ATM Actor: Bank Customer Goal: Withdraw cash from an ATM Steps: Customer inserts ATM card into machine ATM prompts customer to enter 4-digit PIN Customer enters valid PIN ATM displays available account(s) Customer selects account to withdraw from ATM displays available withdrawal amounts Customer selects withdrawal amount ATM dispenses cash ATM prints transaction receipt ATM returns card 3. Book Airline Travel Actor: Traveler Goal: Book flight tickets on an airline website Steps: Traveler visits airline website Traveler selects book flight option Website prompts entry of desired airports and travel dates Traveler enters details Website displays available flight options Traveler selects desired outbound and return flights Traveler provides passenger and payment details Website processes booking and displays confirmation 4. Listen to Music on Streaming Service Actor: User Goal: Listen to specific songs or albums on demand Steps: User opens music streaming mobile app User browses or searches for desired music App displays search results User selects song or album to play App streams selected music User can pause, skip tracks, adjust volume, etc. User can download music for offline listening App recommends related artists and music 5. Search for Information Online Actor: Website user Goal: Find information on a topic online Steps: User opens web browser User navigates to search engine website Website displays search bar User types in keywords for their information query User clicks search button or hits enter Website returns ranked list of search results User browses and clicks on promising results User iterates search to refine results 6. Purchase Insurance Policy Actor: Customer Goal: Buy an insurance policy online Steps: Customer visits insurance company website Customer selects type of policy they want to buy Website asks for some basic personal details Customer provides requested info Website displays quotes for coverage options Customer selects desired quote Website prompts for payment and beneficiary details Customer provides details Website processes payment and issues policy docs 7. Control Smart Home Devices Actor: User Goal: Use a mobile app to control smart home systems Steps: User opens home automation app App displays connected devices organized by room User selects room they want to control User selects specific device like lights or thermostat App displays controls for that device User adjusts device settings like dimming lights or changing temperature App synchronizes updated settings 8. Pay Bills Online Actor: Customer Goal: Pay bills electronically Steps: Customer logs into online bill payment website Website displays customer's pending bills Customer selects which bills to pay Customer enters payment amount for each Customer selects date for each payment Customer submits payment transaction Website processes payments 9. Order Prescription Refill Actor: Patient Goal: Request prescription refill from pharmacy Steps: Patient logs into pharmacy website or mobile app Patient selects prescription refill option System displays patient's available prescriptions Patient selects prescription(s) needing refill System indicates selected prescriptions and refill status Patient confirms refill order Pharmacy processes request 10. Take Online Course Actor: Learner Goal: Take an online course to gain new skills Steps: Learner registers on e-learning platform Learner enrolls in desired course Learner can view course modules and lessons Learner completes lessons which have videos, text, quizzes Platform tracks learner progress and mastery Course provides certificate upon completion 11. Rent a Car Actor: Traveler Goal: Rent a vehicle from a car rental company Steps: Traveler makes reservation on company website Traveler visits rental office to pick up car Agent verifies customer ID and reservation Agent reviews rental terms with customer Customer inspects car exterior and interior Agent records mileage and fuel level Customer signs rental contract Agent provides car keys and directions 12. Withdraw Cash from a Human Teller Actor: Bank Customer Goal: Withdraw cash from teller at bank branch Steps: Customer enters bank branch and gets in teller line Teller prompts customer for ID and transaction Customer requests cash withdrawal amount Teller verifies available funds Teller counts out cash to fulfill request Teller gets customer signature to complete transaction Teller provides customer with transaction receipt 13. Buy Movie Tickets Online Actor: Customer Goal: Purchase movie tickets on cinema website Steps: Customer selects movie theater location Website displays movies and showtimes Customer selects movie, showtime, and number of tickets Customer selects seat assignments from theater map Customer provides payment information Website processes order and displays ticket confirmation 14. Track a Shipment Delivery Actor: Customer Goal: Track status of shipment coming from online retailer Steps: Customer logs into retailer website Customer selects order history and finds relevant order Website displays shipment tracking number Customer copies tracking number Customer visits shipping carrier website 6 Wrike Recognized by TrustRadius for Industry-Leading Usability and Customer Service Wrike earns two awards from TrustRadius for its best-in-class usability and customer service. Find out how to get started with the world's leading digital work hub. The history of the use case Swedish computer scientist Ivar Jacobson presented the first article on use cases in 1987, describing how the technique was used at telecommunications company Ericsson to capture system requirements. In 1992, Jacobson co-authored the book "Object-Oriented Software Engineering – A Use Case Driven Approach," which helped popularize use cases for specifying functional requirements in software development. Jacobson later joined American software engineers Grady Booch and James Rumbaugh to create the Unified Modeling Language (UML), which introduced a standard way to visualize the design of a system. Since then, the technique has been adapted into use case writing "templates" to streamline the capture of high-level requirements. Use Case Description EXAMPLE | Use Case Tutorial and Best Practices | What should be included in a use case? Here are some basic components to include when writing a use case: System: This may include the site, the software, the app or the program you want to run the use case to assess. Including this information is especially important when writing use cases for multiple systems. What is an example of a use case? Use cases often help outline: Use cases often include an index, with several individual use cases for each step of a software or platform's process. For example, use cases about an online e-commerce purchase might include separate reports for adding an item to the cart, selecting a payment type and choosing a shipping and delivery option. How do I write a use case? Writing a use case sounds complex, but only requires understanding your system and its users. You can write a use case by following these six steps: 1. Describe your system Start by describing your system, or the product or service you and your team will build. Focus your description on what your system does for users. What is a business use case? Some teams like to write a business use case to outline a system's processes before development. As developers begin their work, a manager will outline more technical system use cases to follow. Use cases show all the ways a system functions when trying to reach goals, but a scenario only depicts one example.