Industrial Case: Reuse Center

CI Context Viewpoint

1. As-Is Workflow Model

Workflow Name:
Programming PLCs (Programmable Logic Controllers) with the programming language ST (Structured Text)

Domain Context:
The context is software development for industrial automation systems.

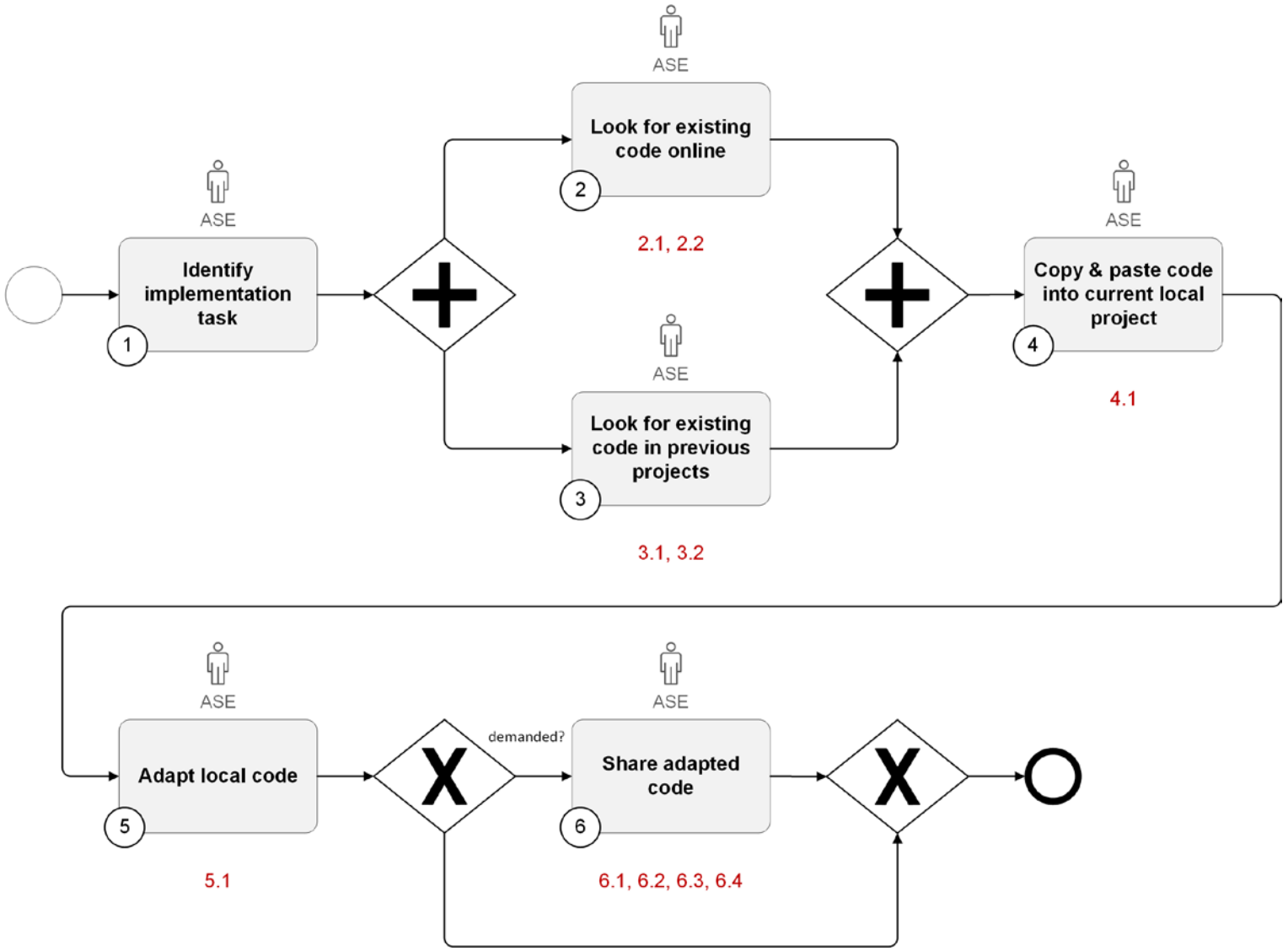
Objective of this workflow:
To provide working automation software solutions.

Stakeholders:

- Automation software engineer (ASE)
  A person who is part of a development team to implement working automation software solutions.
  Their concerns are:
  - An efficient workflow to implement effective and efficient automation software solutions.
  - To be aware about changes in the code base.
  - To create recurring solutions with low effort.
  - An efficient way to maintain software solution.
  - An efficient way to test software solutions.
  - Needs to follow standardized processes and procedures.
  - Sufficient documentation about the implementation of the software solution.

- An efficient way to find best-fitting solutions for a given problem.
- Needs to collaborate with peer ASE developers from the same team and/or from different teams in different business units. Collaboration activities are:
  - working on shared code base,
  - information sharing,
  - plan and design.

Current Business Process:



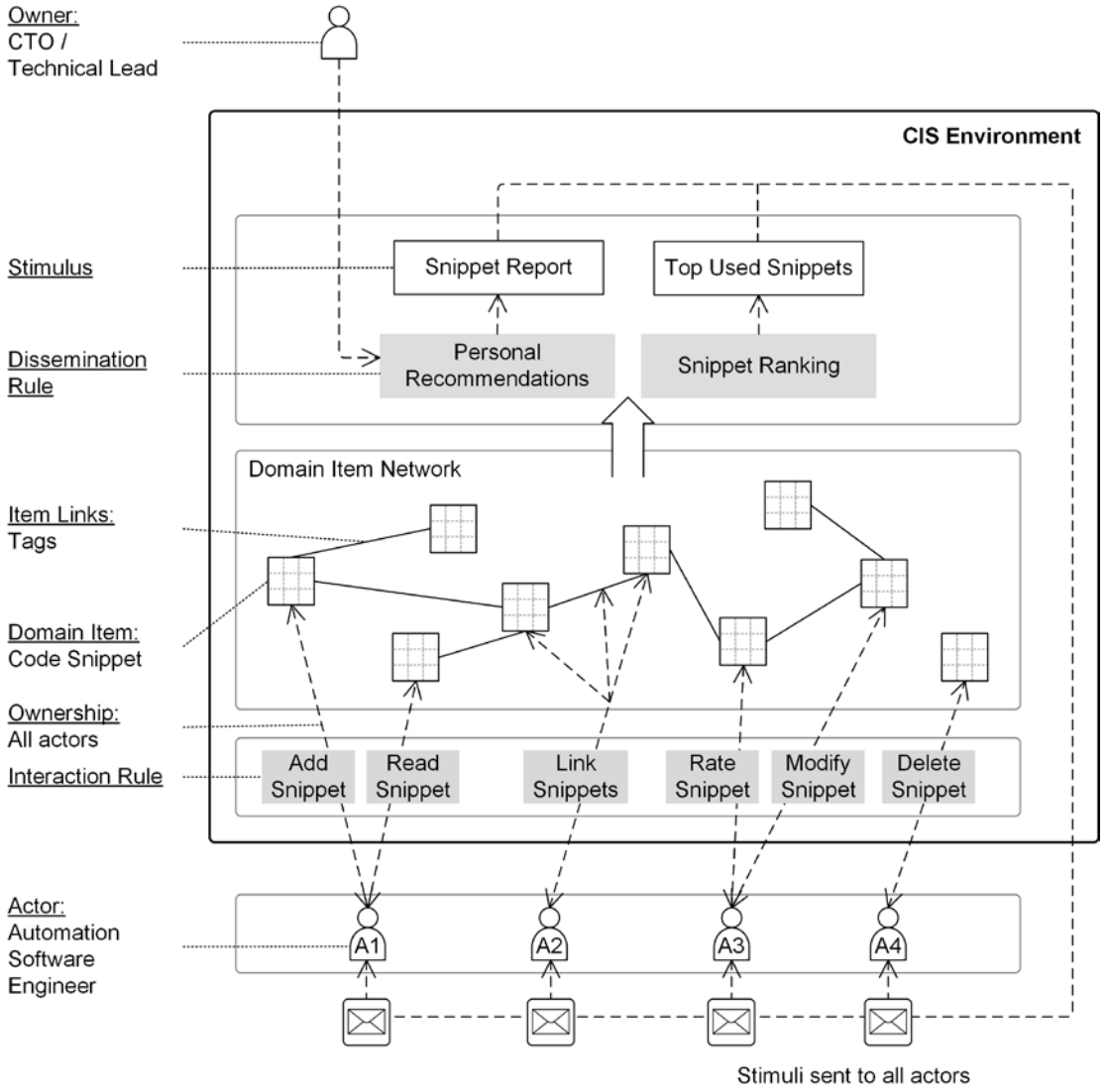
List of Activities:

A#	Activity Name	Activity Description	Stakeholders	Communication Channels
1	Identify implementation task	Each ASE needs to identify his/her implementation task with respect to the given engineering problem.	ASE	local
2	Look for existing code online	To efficiently implement his/her task, the ASE looks for existing code online that could be useful for reuse to solve his/her task.	ASE	wiki, websites, forum
3	Look for existing code in previous projects	To efficiently implement his/her task, the ASE looks also for existing code in previous development projects that could be useful for reuse to solve his/her current task.	ASE	local, f2f, email, phone, messenger
4	Copy & paste code into current local project	If a useful code is found, the ASE copy & paste the snippet into the current local project.	ASE	local
5	Adapt local code	The reused code needs to be adapated regarding the current implementation task.	ASE	local
6	Share adapted code	If demanded, copy & paste adapted code into wiki / forum / email / messenger / etc. to share it with other team members for reuse.	ASE	local, email, messenger, wiki, forum

List of Limitations:

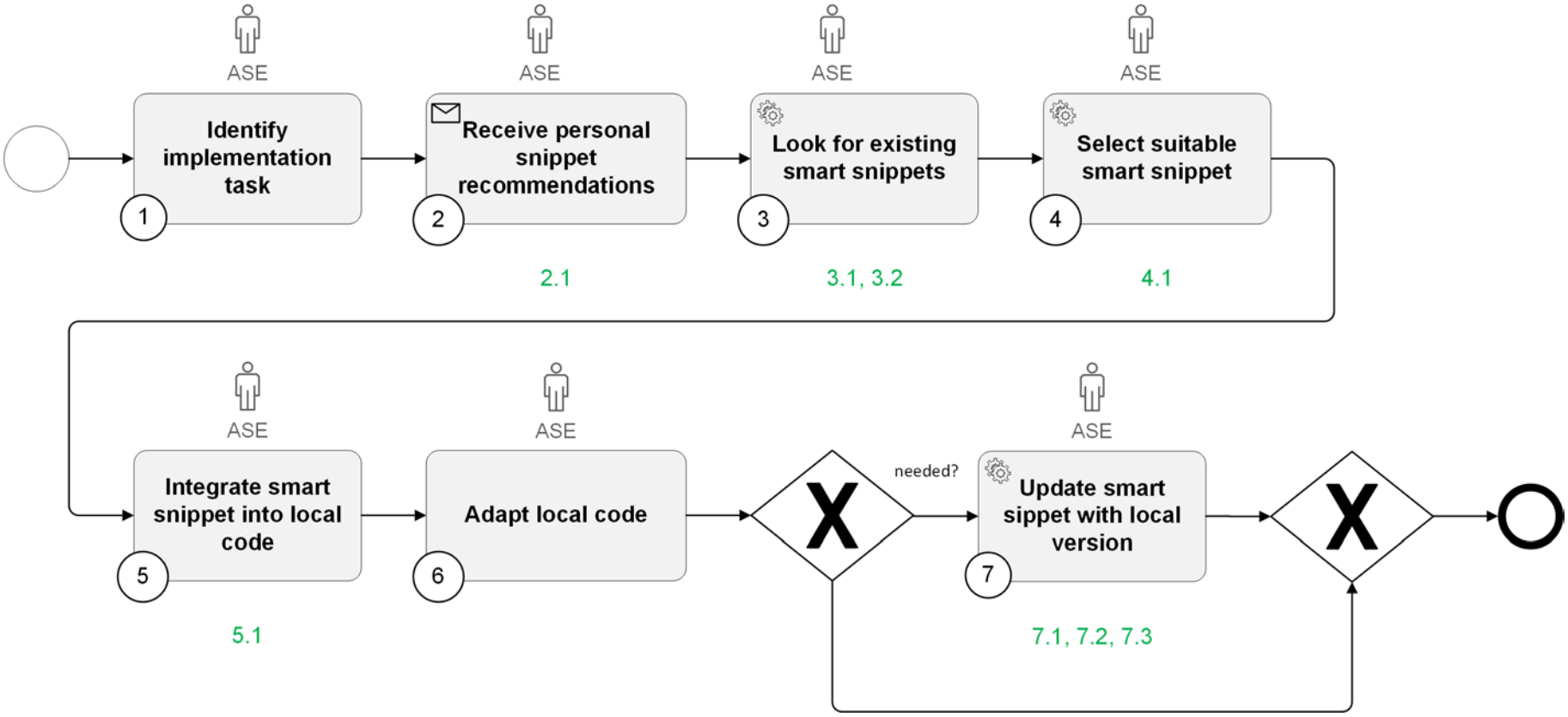
ID	A#	Description	Stakeholders	Impact (1-low, 5-high)
2/3.1	2, 3	Lack of efficiently disoverability of existing and suitable code solutions for reuse. Finding the „best fitting“ solution for a given implementation task is challenging.	ASE	5
2/3.2	2, 3	Knowledge of existing solutions is tacit and sometimes unstructured captured locally by an ASE. Thus this knowledge is hard to access and dispersed across all involved stakeholders.	ASE	4
4.1	4	Lack of tool support to efficiently reuse suitable code for a given implementation task.	ASE	4
5.1	5	Traceability of original source and author is typically lost which complicates back flow of bug fixes.	ASE	2
6.1	6	Quality, structure and completeness of provided code for reuse differs depending on the ASE.	ASE	4
6.2	6	Reusable code is shared with other team members using different tools and services and thus make it more difficult for stakeholders to be aware about existing and suitable code solutions that are available.	ASE	5
6.3	6	The bigger and more locally distributed the development team is, the less effective is the exchange and management of shared code.	ASE	4
6.4	6	Missing support to collaborate to share and clustering code and annotate it with relevant context information.	ASE	3

2. Stigmergic Coordination Model



3. To-Be Workflow Model

Planned Business Process with CIS:



Improvement-Limitation Mapping Table:

ID	A#	Improvent provided by the CIS	Issue (with associated ID)	Agent	Actor Role	System
2.1	2	The RUC regularly notifies actors about personal recommended code snippets that are recently created or modified by others and that could be interesting for the individual actor.	ID 2/3.1: Lack of efficiently disoverability of existing and suitable code solutions for reuse. Finding the „best fitting“ solution for a given implementation task is challenging.			CIS
3.1	3	All actors have access to one shared environment where all shared snippets are captured.	ID 2/3.2: Knowledge of existing solutions is tacit and sometimes unstructured captured locally by an ASE. Thus this knowledge is hard to access and dispersed across all involved stakeholders.	Actor		
3.2	3	A result list of collected and defined code snippets is created implicitly and reflects always the current state of the snippets‘ content which can be accessed by developers across the world via Internet.	ID 6.3: The bigger and more locally distributed the development team is, the less effective is the exchange and management of shared code.	Actor		
4.1	4	The selection of suitable snippet is facilitated by provided context information and clustering mechanisms like tagging which can be updated by each actor.	ID 6.4: Missing support to collaborate to share and clustering code and annotate it with relevant context information.	Actor		
5.1	5	The RUC provides services which can be used by an IDE to have access to the snippet base and to efficiently integrate snippets into local code.	ID 4.1: Lack of tool support to efficiently reuse suitable code for a given implementation task.	Actor		
7.1	7	All snippet modifications are captured as a new version of the snippet in order to enable traceability of all created snippet versions.	ID 5.1: Traceability of original source and author is typically lost which complicates back flow of bug fixes.	Actor		
7.2	7	All snippets are collected and stored in a pre-defined, structured manner to support entry completeness and a common format of data.	ID 6.1: Quality, structure and completeness of provided code for reuse differs depending on the ASE.	Actor		

7.3	7	Actors create and update snippets in one shared space and thereby make them visible to other team members. As a result dispersed versions of reusable code are avoided and the current state of them are available at one location.	ID 6.2: Reusable code is shared with other team members using different tools and services and thus make it more difficult for stakeholders to be aware about existing and suitable code solutions that are available.	Actor		

CI Technical Realization Viewpoint

1. Artifact Definition Model

CI Artifact:
Code Snippet

Artifact Links:

- Tag

Artifact Content:

Attribute Name	Attribute Type	Attribute Description
snippet name	string	unique name
snippet author	string	name of the actor who created the snippet
snippet creation timestamp	timestamp	date and time when the snippet was created in the system
snippet editor	string	name of the actor who last modified the snippet
snippet modification timestamp	timestamp	date and time when the snippet was last modified in the system
snippet description	text	description of the snippet's content
snippet type	String	type of the snippet
snippet content	text	code stored by the snippet
tags	set<string>	tagging enables actors to categorize snippets using their own keywords

revisions	set<snippet>	set of former versions of this snippet's content
discussion	set<comment>	discussion about the snippet
rating	integer	rating of the snippet's quality

Operation Specifications:

- CREATE
- READ
- UPDATE
- DELETE
- COMMENT
- RATE
- LINK

2. Aggregation Model

Actors:

Agent	Actor Role	Actor Role Description	Stakeholder
Actor	*(no specific roles)*		ASE

Actor Record (AR):

Attribute Name	Attribute Type	Attribute Description
actor	string	actor who is the owner of this actor record
actor activity	string	activity performed by the actor and logged by the actor record
snippet	string	snippet on which the actor activity was performed
activity captured timestamp	timestamp	date and time when the actor activity was performed
ip address	string	ip address used by the actor to perfom the activity
user agent	string	user agent used by the actor to perform the activity

List of Actor Activities and Artifact Input/Output Flows:

Client	Type	Activity	Activity Description	Artifact Input	Artifact Output	AR Log
Browser Client, IDE Plugin	W	Create snippet	Add a new code snippet.	*) name (string) *) content (text) *) description (text) *) type (string) *) revision comment (string) *) list of tags (string)		YES
Browser Client, IDE Plugin	W	Modify snippet	Modify metadata of an existing code snippet including updating snippet links.	*) name (string) *) content (text) *) description (text) *) type (string) *) revision comment (string) *) list of tags (string)		YES
Browser Client, IDE Plugin	R	Show snippet	Access a specific snippet and show its details.		*) snippet name (string) *) author (string) *) snippet description (text) *) snippet content (text) *) snippet type (string) *) created_at (timestamp) *) last_modified_at (timestamp) *) latest revision number (integer) *) latest revision comment (string) *) latest revision editor (string) *) average rating (integer) *) list of tags (string) *) list of snippet comments, for each comment: comment author: string, comment text: text, comment_created_at: timestamp)	YES

Browser Client, IDE Plugin	R	Overview all snippets	Access and show a list of all snippets with their metadata from the latest revision.		List of all snippets, for each snippet: *) snippet name (string) *) author (string) *) snippet description (text) *) snippet type (string) *) created_at (timestamp) *) last_modified_at (timestamp) *) latest revision editor (string)	
Browser Client, IDE Plugin	R	Overview of all tags	Access and show a list of related code snippets grouped based on their assigned tags.		List of all tags, for each tag: *) tag name (string) *) list of assigned snippets with their name (string)	
Browser Client, IDE Plugin	W	Delete snippet	Delete an existing code snippet so that it is not available anymore.	*) revision comment		YES
Browser Client	W	Rate snippet	Add a rating to the code snippet to assess its utility/quality.	*) rating (integer: 1-5)		YES
Browser Client, IDE Plugin	R	Show details of a tag	Access a specific tag and show a list of related code snippets that are linked via the tag.		*) tag name (string) *) list of assigned snippets with their name (string), description (text)	
Browser Client	R	Show statistics	Access and show detailed statistics about weakly/monthly activities and active users.		*) List of interesting statistics (monthly/weekly) about snippets, actors	
Browser Client	W	Add comment	Add a new comment to the discussion of a code snippet.	*) comment text (text)		YES

Activity Types: Write Activity (W), Read Activity (R)

3. Dissemination Model

List of Dissemination Rules (push-based):

ID	Trigger	Schedule	Analyzer	Data Source / Filtered Output	Channel	Purpose	Recipient
1	Snippet Report	once a month	*) Recommender System *) Data Mining Component *) Notification Builder	Data Source: *) Actor Records *) Artifact Contents Filtered Output: *) Individually filtered report of recent activities based on tags that the individual actor often reuses (for each activity: actor name, timestamp, term / definition, activity type)	Email	*) Support awareness about news & updates *) Increase engagement level of actors *) Stimulate actors to improve quality of existing snippets and rate their utility	All actors
2	Top Used Snippets Notification	once a month	*) Data Mining Component *) Notification Builder	Data Source: *) Actor Records Filtered Output: *) Ranking of the most used snippets in the last month	Email	*) Support awareness & reputation	All actors

CI Operation Viewpoint

1. Initial Content Acquisition Model

Initial Data:

Artifact Content	Transformation	External Source	Initial Data
*) Snippet Metadata	1) Export of already collected list of snippets from the wiki / repository. 2) Import data set using a function from the RUC's underlying database system which stores the initial data according to the snippet artifact structure.	*) Wiki *) Repository	List of already collected snippets in some format

Initial Actor Profile

- Expertise
  - Automation software engineers with expertise in Structured Text.

- Recruitment
  - A development group / team within an organization who work together in a development project.

2. CI Analytics Model

List of Metrics:

Probe	Time for Measurement	System Component	Property	Metric
Tracker probe	At each load of a snippet's detail view.	Controller component	Actor Activity	Snippet Views by individual actor
Tracker probe	After a new snippet was created.	Controller component	Actor Activity	Snippet Creation by individual actor
Tracker probe	After a snippet's content was modified.	Controller component	Actor Activity	Snippet Modifications by individual actor
Notification probe	At each load of a notification email's content.	Controller component	Actor Activity	Notification Views by individual actor
Notification probe	At each click on a notification email's content that redirects actor to content in the system.	Controller component	Actor Activity	Notification Content Clicks by individual actor
Notification probe	After each notification dispatch to an individual actor.	Mailer component	System Activity	Notification Dispatches to individual actor