■ SCCH ■ Softwarepark 32a ■ A-4232 Hagenberg

SURVEY

THE POTENTIAL OF LOW-CODE DEVELOPMENT IN THE MANUFACTURING INDUSTRY

!!! START RECORDING !!!

PART A: Company, interviewee and system profile

- **Key company data**
  a. Products and/or services
  b. Markets (by location and industry)
  c. Turnover
  d. Number of employees

- **Role and experience of interviewee**
  a. Current and past roles
  b. Level of experience in software engineering
  c. Level of exerpience in low-code software development
     - I have never heard this term before.
     - Limited experience (details)
     - We are employing no-code development platforms at our company (examples)
  d. Area of responsibility

- **Overview of current system**
  a. Architecture (components and connections/interactions)
  b. What are the major implementation units of your system(s)?
     - API
     - Template
     - Component
     - Service
     - Framework
     - Widget
     - SDK
  c. Lines of code, development effort in person-years
  d. Team size and structure (internal, external)
  e. Tech stack (e.g., DSLs)
     - Current system(s)
     - New system (by means of LCDP)

- **Software engineering process** (agile, waterfall, iterative)

PART B: Use cases and requirements

1. **What is your understanding of the low-code ecosystem?**
   a. Open question
   b. Backup disucssion topics

Term	**Example**
No-code, low-code	You need no/less programming skills and you are able to realize your processes without the need of coding.
Drag and drop	Can probably do everything through drag-and-drop.
Visual programming	Visual approach to software development.
Pre-designed templates	Gives every one from business users to advanced developers the right automation canvas to build great software robots.
Non-professional programmers friendly	This is especially useful for people with limited coding skills or devs that want to automate something quickly while not having to think about all aspects of development, such as deployment and security.
What you see is what you get	Meant for WYSIWYG app maker.
Business process	Especially designed for process owners.
Graphical user interface	They provide you with a graphical wizard.
Build automation	Automate unattended operations with minimal human involvement.
Database operation	It provides some cool tools for generating CRUD entities by scaffolding.
Collaboration in the same environment	Combines visual and code workflows allowing designers, developers, and low-code users to work together in a single environment.

2. **(RQ1) Which use cases are you planning on supporting by LCDP? Please rate their payoff vs. difficulty!**
    a. Open Question
    b. Backup disucssion topics

End user robot programming	Data pipeline implementation (low-code AI Development)
End user HMI customization	Data analysis and visualization, dashboards
HMI order development	Data platforms, IIoT hubs
Implementation of supportive systems	Digital twin development
Machine control development (PLC, µC, NC)	AR/VR/XR

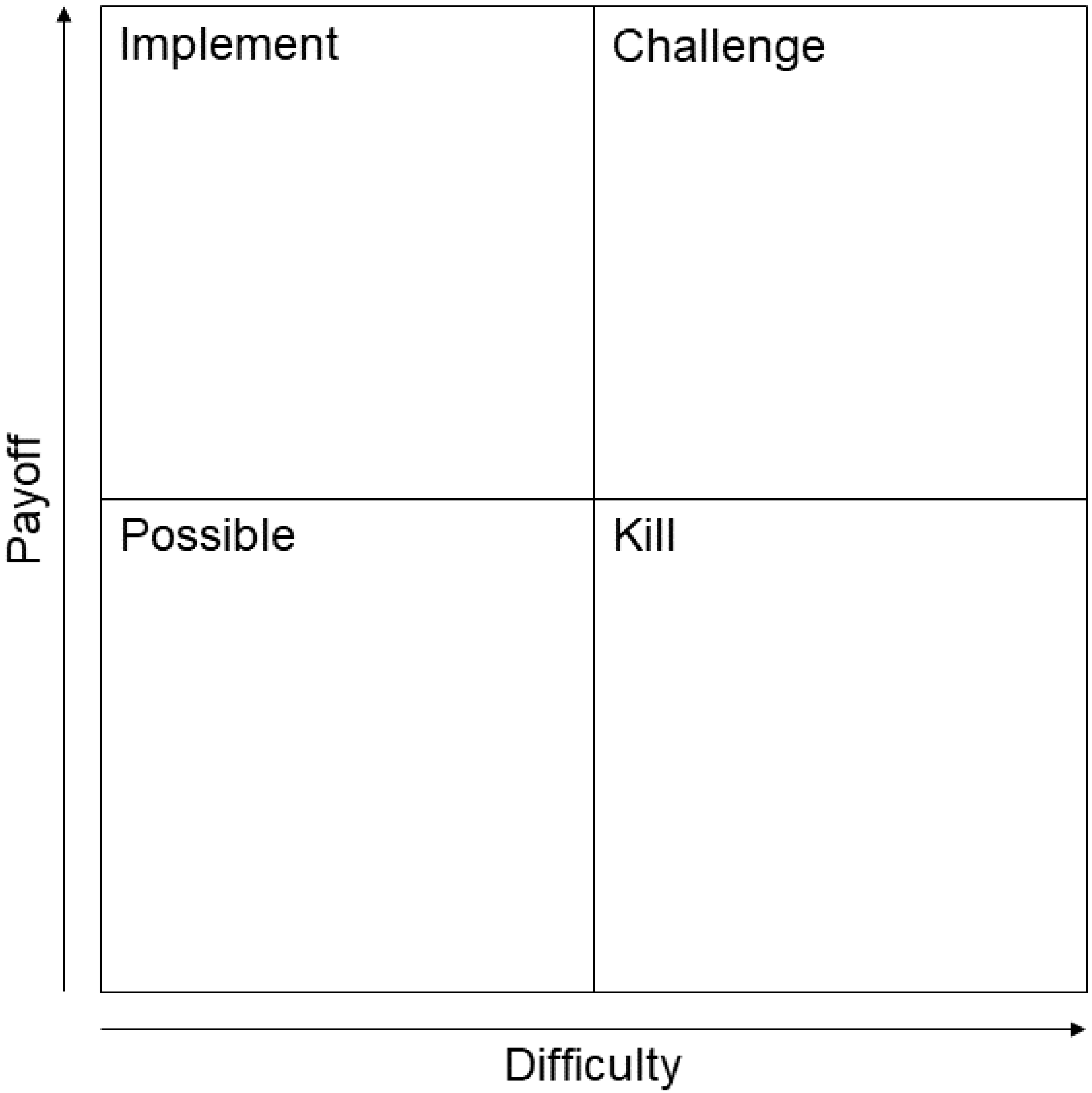
A F

B G

C H

D I

E J



Payoff

Implement

Challenge

Possible

Kill

Difficulty

A …

B …

C …

D …

E …

F …

G …

H …

I …

J …

3. **(RQ2) What are the resulting (high-level) requirements for LCDPs that support these uses cases?**

   a. Open question

   b. Backup disucission topics

Who is the low-code programmer? Client/customer or manufacturer?	Communication/compatibility with external/vendor systems • MES • PLC IDE • OPC UA
Graphical user interface	Interoperability support
Security support	Collaborative development support
Reusability support	Scalability
Business logic specification mechanisms	Application build mechanisms
Deployment support	

PART C: Benefits, limitations and challenges

1. **(RQ3) What benefits do you expect?**

Benefit	**++**	**+**	**-**	**--**	**N/A**
Faster development	○	○	○	○	○
Ease of study and use	○	○	○	○	○
Lower IT costs	○	○	○	○	○
Rich and ready-to-use units	○	○	○	○	○
Newbie friendly	○	○	○	○	○
Improved system quality	○	○	○	○	○
Strong integration and expansion capability	○	○	○	○	○
Minimal effort	○	○	○	○	○
Better customization	○	○	○	○	○
Perceptual intuition	○	○	○	○	○
Superior usability	○	○	○	○	○
Better user experience	○	○	○	○	○
Easy deployment	○	○	○	○	○
Cost-effectiveness	○	○	○	○	○
Improved IT governance	○	○	○	○	○
Improved team development	○	○	○	○	○

2. **(RQ3) What limitations and challenges do you anticipate?**

Limitation/challenge	**++**	**+**	**-**	**--**	**N/A**
High learning curve	○	○	○	○	○
High pricing	○	○	○	○	○
Lack of customization	○	○	○	○	○
Slow loading and publishing	○	○	○	○	○
Less powerful than programming	○	○	○	○	○
High complexity	○	○	○	○	○
Complex issues still need coding	○	○	○	○	○
No access of source code	○	○	○	○	○
Not really ease of use	○	○	○	○	○
Limitation to experienced developers	○	○	○	○	○
Vendor lock-in	○	○	○	○	○
Difficulty of maintenance and debugging	○	○	○	○	○
Difficulty of integration	○	○	○	○	○
Unfriendly user experience	○	○	○	○	○
Need of basic programming knowledge	○	○	○	○	○

++	**+**	**-**	**--**
Strongly agree	Agree	Disagree	Strongly disagree