Citizen Developers Driving the Digital Campus

Christoph Baumgarten[1], Alex Simeon[2], Michael C. Wilhelm[3]

[1]FHS St. Gallen, University of Applied Sciences, Switzerland, christoph.baumgarten@ost.ch
[2]HSR University of Applied Sciences Rapperswil, Switzerland, alex.simeon@ost.ch
[3]NTB University of Applied Sciences Buchs, Switzerland, michael.wilhelm@ost.ch

Keywords
Digital transformation, digital campus, learning organization, citizen development, low-code platforms, PowerApps, oauth2.

1. ABSTRACT

What could a university look like that has successfully entered the digital age? This question has been intensively discussed at three Swiss universities of applied sciences, which are currently in a merger process. The result is a holistic architecture for a *digital campus* that not only covers infrastructural aspects, but also contains content-, skills-, management- and culture-related building blocks for a comprehensively digitalized university.

The deliberate active involvement of campus citizens as so-called *citizen developers* by guiding and supporting their hands-on participation in the university's digital transformation efforts is identified as an important leverage for implementing the digital campus and illustrated by an example.

2. DIGITAL TRANSFORMATION CHALLENGING UNIVERSITIES IN TWO DIMENSIONS

There is hardly any other megatrend that currently affects society, companies, institutions and individuals as comprehensively as the digital transformation (OECD, 2017). Dealing with the raising opportunities and risks is prior - a development that is of course leaving its mark also in higher education: Curricula in practically all disciplines are expected to take the cross-sectional topic of digitalization[1] appropriately into account, and the universities are doing their utmost to meet this existing demand.

Actually, universities are *challenged in two dimensions* with regard to the digital transformation:

(1) *What* the university transports to the outside world, namely domain-specific teaching content, research results, consultations, etc., should take the topic of digital transformation with its respective technical, organizational, legal and cultural aspects adequately and well balanced into account. For example, a lecture could pay too little attention to the cultural effects of the digital transformation, but pay too much attention to technical aspects, or serve diffuse cultural pessimisms, have too little practical relevance, etc.

(2) *How* a university operates internally and at its interfaces, i.e. in its processes and services, should keep pace with an increasingly competitive environment, while ensuring that objectives resulting from the university's mandate are best achieved, and cost are kept at reasonable levels. To achieve this, the university will have to engage in the digitalization of its own organization and enter itself into a fundamental transformation process – an argument supported by numerous studies and no longer much debated today; see for example (Licka & Gautschi, 2017).

To successfully cope with these two challenges, universities must leave their traditional and well-ordered gardens: "Digitalization is not a technology, but a mental attitude" (Bredlow, 2019), it is a shift in culture, a shift in the "set of values, norms, practices, and expectations" shared within the organization (Deuze, 2006) (Hemerling, Kilmann, Danoesastro, Stutts, & Ahern, 2018). Such a *digital culture* "empowers people" and "attracts talents", it

- encourages trying out ("fail fast, fail often, but learn"),
- promotes the view from the outside to the own services ("how does a *student* perceive our study offerings?"),
- values information sharing as well as joint efforts (one-man-shows and information silos slow

[1] The term 'digitalization' is used as a synonym for 'digital transformation'.

down almost every organization), and
- avoids over-planning and explicit top-down steering (better rely on iterative approaches and guiding principles rather than on detailed strategies, commands and control).

Thus, tackling the two challenges means changing the very fundamentals; a proceeding will therefore be stepwise: Universities will have to understand the phenomena of digitalization, integrate it into their curriculars and apply it to their own organizations. They will then analyze what has been achieved, hence reflecting digitalization in their various contexts, and derive further action.

Overall, they face a typical double loop learning scenario (Argyris, 1991), whereas the object of study is, due to the particular nature of educational providers, considered twice: digitalization integrates into curricular content as well as research, and at the same time is paradigm to the organizational change of the universities.

Therefore, the universities' learning process in addressing the two challenges can benefit from an additional, self-reinforcing cycle, compare Figure 1:

- To further develop and complete "what the university transports to the outside world" with regard to digitalization – challenge (1) -, opportunities for academic personnel and students should be created that allow for a gathering of practical know-how in digital transformation. This follows the logic of *makerspaces* – environments used for developing ideas and constructing them into some tangible form. The process of constructing, the iterative transformation of an idea into a concrete, possibly physical or digital representation, "supports learners' conceptual understanding" (Sheridan, et al., 2014). It allows learners to experience which leads to an active knowledge building. Compare the theories of *constructivism* and *constructionism* (Papert, 1993) (Cross, 2011).
- Opportunities to gather practical know-how in the field of digitalization could be taken from the stock of ideas and measures emerging from challenge (2). This way, the academic personnel und the students contribute with their insights, but also their manpower, to the digital development of "how the university operates".

'Not just talking about digital transformation but living it' could be the mantra of such a proceeding, which is further elaborated in Section 4.

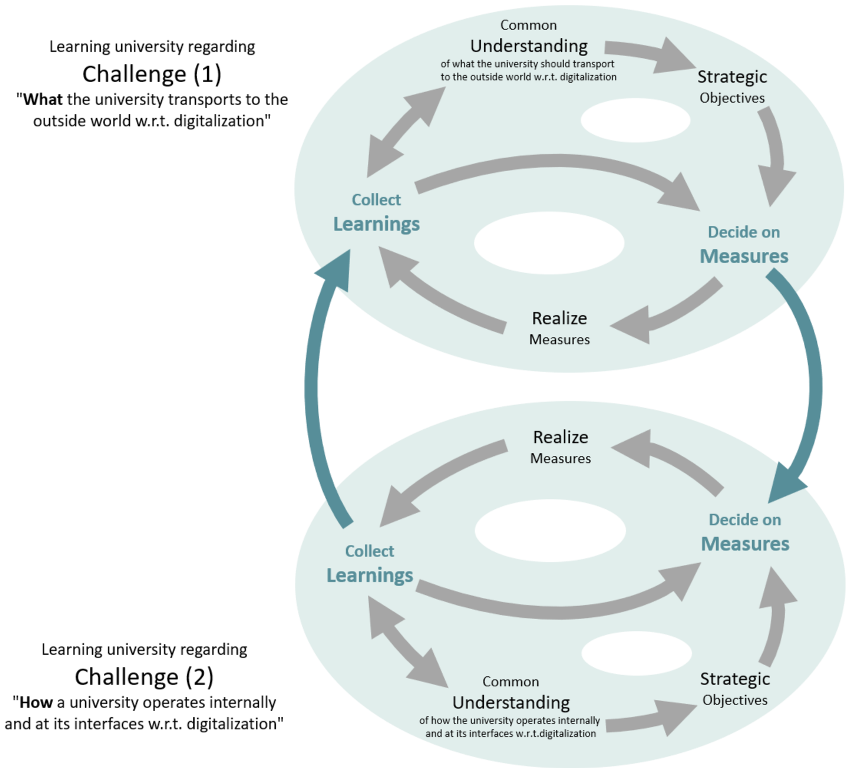


Figure 1: The two challenges that a university in digital change faces each imply a double loop learning, which can be linked so that they are mutually reinforcing

3. A HOLISTIC COMPREHENSION OF THE DIGITAL CAMPUS

Starting from the two challenges discussed above for a university on its journey to digital transformation, a range of objectives can be collected tackling these two challenges. Table 1 summarizes a selection of such objectives and lists corresponding components of an architecture implementing a digital campus according to the objectives. Assembling the derived components leads to Figure 2 illustrating the *holistic architecture of a digital campus*.

Table 1: Selected objectives and components of a digital campus

<table>
<tr><th colspan="2">Challenge</th><th>Objective</th><th colspan="3">Addressed by architecture component</th></tr>
<tr><td rowspan="2">(1) what</td><td rowspan="2">Adequate balanced covering of the cross sectional topic of digitalization in curriculum and research</td><td>Setup systematic review of the curriculum and research focus to adjust them if necessary (i.e. extend existing processes)</td><td colspan="2">Curricular Content on Digitalization, Digitalization in Research</td><td rowspan="8">Governance, Digital Culture, Digital Codex</td></tr>
<tr><td>Create opportunities for academic personnel and students to gather practical know-how in digital transformation; compare sections 2 and 4</td><td colspan="2">Digital Competences of Students and Academic Personnel</td></tr>
<tr><td rowspan="7">(2) how</td><td rowspan="2">Learning objectives are best achieved</td><td>On a physically distributed campus, establish state-of-the-art teaching and learning methods utilizing digital means</td><td>Digital Teaching & Learning</td><td rowspan="2">Digital Media</td></tr>
<tr><td>Ensure Learning Success</td><td>Digital Control & Support Systems</td></tr>
<tr><td rowspan="3">‘Customer journeys’ of (prospective) students, academic personnel etc. should meet expectations</td><td>Provision of an end-to-end digitalization of the essential learning, teaching, research and administrative processes tailored to occasional users</td><td colspan="2" rowspan="2">Digital Services,
Digital Competences Staff</td></tr>
<tr><td>Ensure that the university is optimally perceived by the outside world, so that its reputation gradually improves</td></tr>
<tr><td>Establish culture and digital means for a highly effective collaboration</td><td colspan="2">Digital Knowledge Communities</td></tr>
<tr><td>Efficient administration</td><td>Provide end-to-end digitalization of administrative processes tailored for power users</td><td colspan="2">Digital University Administration</td></tr>
<tr><td colspan="2">The necessary infrastructural basis exists</td><td colspan="2">Master and Semester-related Data & Operations, Facility/IT Infrastructure</td></tr>
</table>

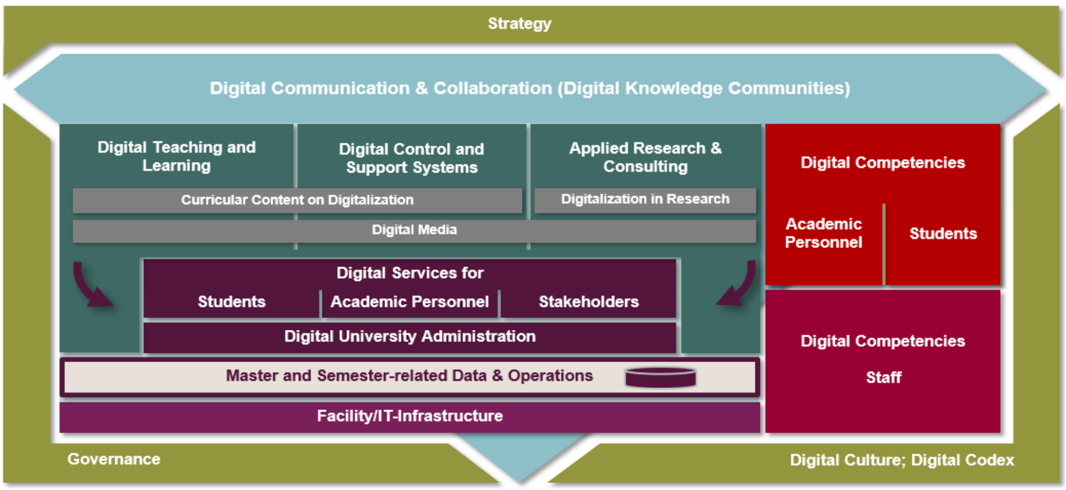
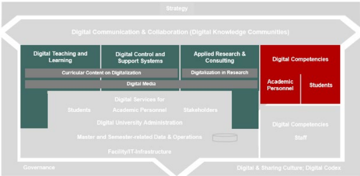


Figure 2: Holistic architecture of a digital campus

Challenges, objectives and architecture were derived from a series of workshops on the design of a joint digital campus, which were conducted by selected delegates from the three currently merging universities of applied sciences, Buchs, Rapperswil and St.Gallen, in eastern Switzerland (Ost - Eastern Switzerland University of Applied Sciences, 2020). Each of these delegates maintained a feedback loop in their respective organizations (blogs, info meetings, etc.) to reflect on the workshops’ intermediate results and to prepare for the follow-ups.

In the following, selected components of the architecture are described in more detail:



A university usually runs four core processes - it provides teaching services at bachelor, master and doctoral level, it supplies services in continuing education, it excels in research, and finally offers consulting for economy and society. The presented architecture implements these processes in various digital-supported components:

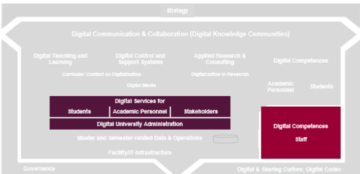
The component *Digital Teaching & Learning* covers all capabilities necessary for digitally enhanced, possibly cross-locational teaching and learning setups. These setups will of course rely on *Digital Media*, which are produced with the help of accompanied media production environments. Example setups include blended learning (Bonk & Graham, 2012) or flipped classroom approaches (Tucker, 2012), but also more unconventional approaches such as topic-specific learning labs. Training factories fall into this category, as do learning labs that simulate demanding care scenarios as part of a nursing education.

The component *Digital Control & Support Systems* provides the means for IT-based evaluations. Moreover, it contributes to a learning analytics based on methods of artificial intelligence.

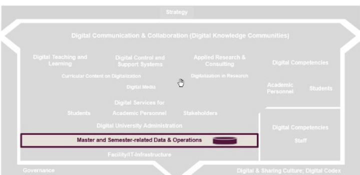
Applied Research & Consulting comprises the digital information supply, the necessary means to effectively network, acquire and publish, appropriate digital-supported research laboratories, and so on.

The components *Digital Teaching & Learning* as well as *Applied Research & Consulting* in particular are also linked to the question of what, in respect to digitalization, is being transported to the outside world – the process of frequently reviewing the alignment of study programs and research portfolios needs to be extended to include a verification of the university-strategic appropriateness and completeness of the actual *Curricular Content on Digitalization* and the actual *Research on Digitalization*.

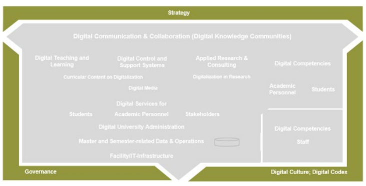
In order to establish all these components, the formation of corresponding *Digital Competencies* with the *Academic Personnel* is indispensable. Section 4 illustrates a way to accelerate this forming. Operating these components will establish the desired Digital Competences on the *Students'* side.



The *Digital Services* provide all functionality commonly needed by students and academic as well as other staff to administrate their study, teaching, research and other business; they are supplied out of the university's back-office, the *Digital University Administration*. These services are offered in the form of integrated mobile one-stop-shops, well supporting *occasional users* (not power users), implemented end-to-end, and follow an omni-channel customer relationship management (CRM) philosophy supporting the entire life cycles of enrolled students, lecturers, prospective students, alumni, etc. They integrate with learning platforms, enterprise content management systems (CMS), etc. Overall, they are designed to effectively fulfill expectations influenced by increasingly inspiring off-campus user experiences. In order to develop these services, the establishment of corresponding *Digital Competencies* in organizational development and technology on the administrative *Staff* side is mandatory.



Part of the architecture is an access-secured application programming interface (API) as a managed shared service providing the essential *Master and Semester-related Data* from a central source of defined quality, as well as *Operations* on the data (Stone, 2019). It serves the applications of the central Digital University Administration and Digital Services for students, academic personnel and other stakeholders. In addition, it is part of the technological basis for scaling the implementation of the digital campus described in Section 4.



A *Strategy*, defined *Governance*, *Digital Culture* and *Codex* form the framework of the digital campus. In the following, a couple of aspects are exemplarily listed that illustrate the broad range of considerations to be taken into account in this context:

A focused implementation of the digital campus requires "strategic guard rails" that keep the involved aligned, i.e. statements determining for example the targeted adoption rate of technological innovations (“early mover” vs. “follower”), the basis for decisions on individual software developments (“make or buy”) or the degree of infrastructure and data security to be achieved (safety and security goals, data classification).

The digital campus thrives on the initiative and dynamism of its actors, their willingness to experiment, short ways of decision-making and intense networking. It is important to find the right balance between a dynamic, i.e. decentralized approach to decision-making on the one hand and the enforcement of guidelines and standards, for example as a result of legal requirements, on the other. Based on the principles of a digital culture, employees should be empowered to take more decisions rather than less.

To what extent does the role of the university's central IT department change when the university enters its digital transformation? Some guidance on this question can be found in Section 4.

The promotion of a sharing culture, transparency, increased exchange of knowledge and information through digital communication and collaboration tools is essential - an honest "together we are strong" attitude of all university members would be very helpful in this context. The university management should actively demonstrate corresponding behavioral patterns, and reconsider how to deal with the internal competition typical of universities, which might oppose a culture of sharing. Answers must also be formulated to occasionally occurring bureaucratic reflexes and lack of pragmatism.

Today's students mainly belong to the generation of the so-called *Millennials*, i.e. those born around the turn of the millennium: Due to their socio-historical background, topics such as globalization, the Internet, social media or the digitally supported individualization of offers no longer require explanation for millennials. In contrast to most of the university's academic and other staff, they are "digital natives", but sometimes find it difficult to put themselves in the shoes of "digital immigrants". Here, appropriate measures are needed to ensure optimal bi-directional learning from each other with regard to the topic of digitalization and the digital transformation of the university, the latter finally having to be carried out in unison by both staff and students (Johnson, 2017).

4. SCALING THE IMPLEMENTATION OF THE DIGITAL CAMPUS

Creating opportunities for academic personnel and students to gather practical know-how in digital transformation – this course of action has been motivated in Section 2 und listed as an objective related to challenge (1) in Section 3. But for these *campus citizens*, how could such opportunities look like?

Enabling opportunities *on campus* for a hands-on, eventually experimental approach to digitalization and taking them from the stock of digitalization ideas implied by challenge (2) is proposed as *one* possible measure: Interested or selected campus citizens could be empowered to digitalize their daily knowledge work, teamwork, courses, research or administration processes, to digitally experiment in class, and so on. In other words, campus citizens cover parts of the digitalization of their working environment and thus learn to better understand the concept of digital transformation - their own digital campus as a *lab for the digitalization of knowledge organizations*.

What is utilized here is the concept of *citizen development* (Everhard, 2019) (Stone, 2019): A citizen developer is a *user* usually with no particular ICT background who creates applications for consumption by others (and of course by her- or himself). Citizen development might employ the same development and runtime environments as the regular professional software development, but this would require correspondingly profound know-how on the side of the citizen developer. This is where *low-code* platforms come into play, that are sufficiently applicable to citizen development, whereby the term ‘low-code’ has been introduced by Forrester (Richardson & Rymer, 2014). These platforms allow citizen developers to generate business applications with minimum coding efforts to express business logics and frontend designs, as well as low integration and deployment efforts to put the application into operation and provide is to other users. All this is achieved by platform features such as

- a development environment with visual (model) editors to generate user interfaces, interpretable or executable models and code, respectively;
- connectors to conveniently access databases, to integrate system interfaces providing data or operations;
- built-in security through encrypted data transfer, integration of identity providers, role-based access control, etc.;
- versioning and continuous deployment;

- code or model interpreting runtime environments, runtime services as logging or scheduling;

and so on. The low-code approach adopts concepts known as *rapid application development (RAD)* (Beynon-Davies, Carne, Mackay, & Tudhope, 1999) or *model-driven software development (MDSD)* (Völter, Stahl, Bettin, Haase, & Helsen, 2013) and embeds them into a full application lifecycle support, low-code platforms are often offered as cloud-based services (*PaaS*). Microsoft's Office/ Dynamics365 with PowerApps, Flow, etc. is one example, others are Mendix or Outsystems. Market overviews are for example given in (Iijima & others, 2019) and (Richardson & Rymer, 2016).

The use of a low-code platform within an organization should be *accepted* and *supported* by the organization's central IT[2]; citizen development should not be misunderstood as an unguided 'shadow IT'. IT should "embrace" (rather than battle) citizen developers and provide the low-code platform as a digital service, on which citizen development can build upon, complemented by self-services, training, support, the promotion of user communities and some indispensable rules to remain secure and compliant (e.g. define and control the responsibilities of citizen developers, regulate the provision of access to applications and data). In addition, IT should be ready to take over the maintenance of selected citizen-developed solutions that have proven their value in practice and should therefore be extended to wider parts of the organization. As a whole, IT should become the *lever* for a broader digitalization campaign (Stone, 2019).

Thus, by enabling and training a university's campus citizens through the central IT to take over parts of the university's digitalization efforts, they are becoming citizen developers. Together with the low-code service provided by central IT, they use the API introduced in Section 3 providing master and semester-related data and operations. This API is pre-integrated by the central IT into the low-code environment in a way such that the actual use of the API by the citizen developer can be easily accomplished.

What must be clearly stated at this point: Low-code approaches have their limitations. There will always be requirements that cannot be reasonably realized with a low-code platform. These cases have to be addressed with classical regular software development by professional software engineers.

A proof-of-concept (PoC) is currently underway, with selected accompanied campus citizen developers employing PowerApps and accessing an API prototype access-controlled on the basis of the oauth2 standard (Parecki, 2012) that provides selected course- and student-related data. In the following, a very first outcome is exemplarily described: a digital solution created by a campus citizen in the role of a citizen developer that supports a specific teaching scenario. This example illustrates aspects like:

- What might be a typical scenario (of moderate scale) in the higher education context that can be effectively improved through citizen development?
- How could the setting for a citizen development be designed concretely, which development environment could be provided? Which kinds of developmental complexity must a citizen developer be able to handle himself, to what extent can complexity be hidden from the citizen developer?
- What typical learnings does a citizen developer draw from his practical digitalization activities; what learnings might show up at the organizational level?

4.1. An exemplary citizen-developed solution

The considered citizen developer is acting as a lecturer in the field of economics. Part of one of his courses is a sequence of moderated discussions on various previously announced topics. Students prepare for these discussions, their contributions are rated immediately by the lecturer and a second assisting co-lecturer: Valuable answers receive two points, all others lead to a deduction by one. Both lecturers used to collect their ratings on prepared paper forms. This data was then transferred into Microsoft Excel for final aggregation.

To avoid the tedious and error-prone work of collecting and preparing scores, the lecturer, as a citizen developer, decides to create a tablet-device-compatible app for capturing these ratings during the discussion. This *rating app* shows the fixed seating order: one touchable and labeled button element per student, see Figure 3. When a particular student is selected for contribution, lecturers press the

[2] A central IT might use such a platform also by itself, to accelerate its own development speed.

button representing this student. The app registers the student by setting the student's rating to -1. If the student has finished and a lecturer considers what he has heard to be valuable, he presses the student's button again, which raises his rating for the student from -1 to 2. Clicking the rating button of another student will register that student, i.e. that student's rating is set to -1, and so on[3]. All ratings are accumulated in a list that can be analyzed afterwards by the lecturers.

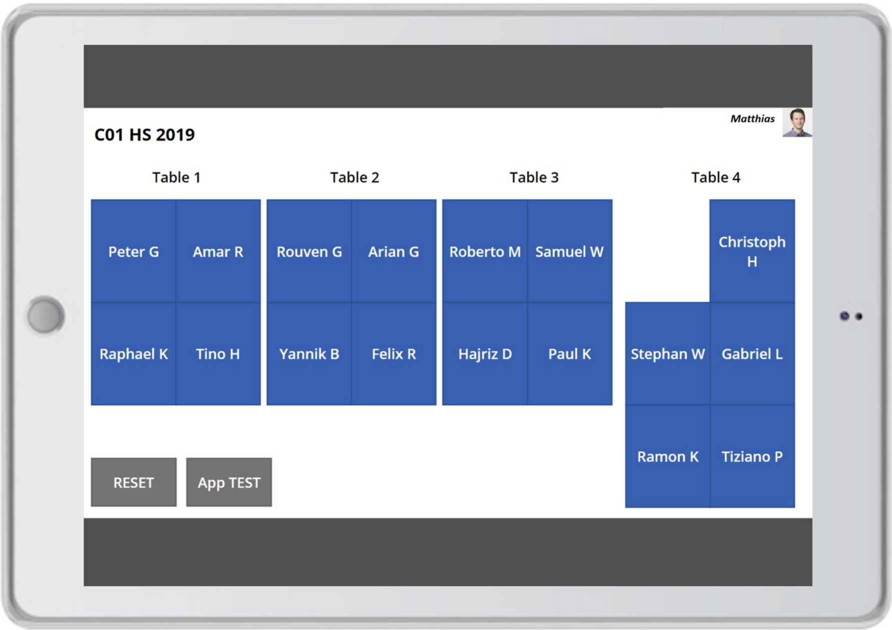


Figure 3: The rating app, citizen-developed using the PowerApps low-code platform, Sharepoint as the app's "local" store, and an access-protected API

The PowerApps-based app created by the citizen developer retrieves necessary student-related data from an API prototype that provides some selected course- and student-related data; it is wrapped into a so-called custom connector that has been shared with all users of the app. Due to this wrapping approach, the citizen developer does not have to cope with the details of an oauth2-based accessing of the access-controlled API – by sharing the custom connector, the rights linked to the custom connector accessing the API are shared as well.

Once the app has loaded the list of students attending the considered course via API using the custom connector, this data is transferred into a Microsoft Sharepoint list "local" to the app, holding the accumulated ratings per student. Furthermore, suitable labels of the rating buttons are derived out of the loaded student data, see Figure 4 illustrating the corresponding PowerApps formula code. Figure 5 shows the formula code executed when clicking a rating button; both figures together illustrate pretty much the entire code written by the citizen developer to define the app's behavior. Note that PowerApps supports code development with sophisticated auto-complete functionality.

As a one-time manual effort at the beginning of the course, enough evaluation buttons must be created by the citizen developer, numbered and dragged to correctly reproduce the seating arrangement. If the course is repeated, the app can be reused; any missing or superfluous buttons must be added or removed (in case there are more or less participants than in the previous course), buttons have to be rearranged to reflect the new seating order.

The following preparing and supporting measures were taken to pave the path for the citizen developer. The subject areas touched here should, in a productive setup, be covered by a central IT:

- The oauth2-based access controlling for the RESTful API prototype that is hosted in the Microsoft Azure Cloud needed to be put in place (Fielding, 2000); the Azure Active Directory (AAD) is used as authentication provider and authorization broker. To simplify the generation

[3] If, and this is a rare case, a registered student is to be registered right away for a second time, the registration can be renewed using a dedicated reset button.

of the custom connector wrapping the API, an OpenAPI specification was prepared that describes the structure of the API including available endpoints with their operations and the necessary security definitions (Swagger, 2016), see Figure 6. The API as well as the custom connector, both had to be registered as apps with the used AAD.

- The citizen developer was assisted by an experienced PowerApps user who introduced the fundamentals of the object-oriented coding approach used in PowerApps and provided helpful hints for handling the PowerApps platform, in particular also regarding the platform's limitations.

```
Set(selectedStudentId, "");                          // Variable containing the id of the currently rated student

ClearCollect(Students,
  testStdAdmAPI.getStudentsFromCourse("HS2019", "C01"));
                                                     // Calls the custom connector wrapping the api, puts the student
                                                     // data related to course C01/HS2019 into the collection variable
                                                     // Students

ForAll(Students,                                     // Loops through Students, synchronizes the list Rates of
  If(                                                // accumulated ratings per student, that is stored in Sharepoint
    CountIf(Rates,
      StudentId = id && Lecturer = User().FullName) = 0,
    Collect (Rates,
      {StudentId: id, Points: 0, Lecturer: User().FullName}
  ))
);

Set(Id1,                                             // For each button, set two variables for convenience reasons...
  Last(FirstN(Students, 1)).id
);
Set(Name1,
  Concatenate(
    Last(FirstN(Students, 1)).firstname, " ",
    Last(FirstN(Students, 1)).lastname
  )
);

Set(Id2, ...
...
```

Figure 4: PowerApps formula code interpreted when starting up the rating app

```
If(
  selectedStudentId <> Id1,
    UpdateIf(Rates, StudentId = Id1 && Lecturer = User().FullName, {Points: Points - 1});Set(selectedStudentId, Id1),
                                                     // Set rate for student (here student related to Id1) to -1
                                                     // and selectedStudentId to Id1, if student's rating button
                                                     // has been hit the first time
    UpdateIf(Rates, StudentId = Id1 && Lecturer = User().FullName, {Points: Points + 3});Set(selectedStudentId, "")
                                                     // Increase rate for student by 3 and unset selectedStudentId,
                                                     // if student's rating button has been hit a second time
)
```

Figure 5: PowerApps formula code interpreted when clicking rating button 1. Analogue code fragments are attached to the other buttons.

Finally, some of the feedback is listed given by the campus citizen who was acting as a citizen developer:

- "The rating app is useful for me and my co-lecturer; it will help us repeatedly save a bit of time. We used it in a first trial, and it worked pretty well!"
- "Would not have succeeded in setting up the app without some initial support. In the end however, I'd consider the complexity to assemble an app with PowerApps be comparable to the setup of an advanced Excel sheet."
- "While reflecting on the design of the tool, a number of ideas came in my mind how to improve the rating process itself. The procedure could become more reliable, for example, if there were an indication as to which of the students have had a small number of contributions so far - I would give them preference in my selection. The indication could be given by the app, by coloring the corresponding rating buttons. Another idea: Build a second app for students that they can use to right away check their results – so far, students get no intermediate feedback during the course."
- "I also see a number of points on how the current app could be further improved: Making the app more flexible so it could easily be used in other semesters or other courses. Or automate

the process of converting achieved points into grades and write the grades back to the central study administration via the custom connector, and so on."

- "First teaching colleagues are interested in that rating app and could imagine of similar applications in their contexts."

```
{
  "swagger": "2.0",
  "info": {
    "version": "2.0",
    "title": "testStdAdmAPI",
    "description": "Test API providing first functionality of a student administration",
    "contact": {
    "name": "citizendev1",
    "email": "citizendev1@ostipm01.onmicrosoft.com"
    }
  },
  "host": "teststdadm-ostipm01.azurewebsites.net",
  "basePath": "/",
  "schemes": [
    "https"
  ],
  "securityDefinitions": {
    "AAD": {
      "type": "oauth2",
      "flow": "implicit",
      "tokenUrl" : "https://login.microsoftonline.com/6002f1ae-9af5-4c37-995b-daca677ffd37/oauth2/v2.0/token",
      "scopes": {}
    }
  },
  "paths": {
    ...
    "/courses/{semester}/{cname}/getstudents": {
      "get": {
        "summary": "Get students attending course in semester",
        "operationId": "GetStudentsFromCourse",
        "produces": [
          "application/json"
        ],
        "parameters": [
          {"type": "string", "description": "semester label", "name": "semester", "in": "path", "required": true},
          {"type": "string", "description": "course name", "name": "cname", "in": "path", "required": true}
        ],
        "responses": {
          "200": {
            "description": "OK",
            "schema": {
              "type": "array",
              "items": {
                "type": "object",
                "properties": {
                  "firstname": { "type": "string" },
                  "lastname": { "type": "string" },
                  "id": { "type": "string", "description": "Student ID", "example": "2017-442" }
                }
              }
            }
          },
          "400": {
            "description": "Bad Request"
          },
          "500": {
            "description": "Internal Server Error"
          },
          "default": {
            "description": "Operation Failed."
          }
...
```

the title will be used to label the custom connector and to access the api object in PowerApps

where the api is hosted

the neccessary security definitions

One of the api's endpoints, providing the operation to retrieve the list of students attending a course related to a certain semester.

the operationId is used to access the operation in PowerApps

the two path parameters of the operation

the type definition of the response of the operation, when the operation request could be successfully processed

Figure 6: Excerpt of the OpenAPI specification of the API prototype that provides some first course- and student-related data

In guided sessions as part of the PoC, the experiences and findings of the PoC's participants such as those listed above are fitted into the concept of digital transformation. This way, the participants gain a better understanding of digital transformation in general.

4.2. Observing the learning university in its digital transformation

The above listed citizen developer's remarks and his participation to the mentioned guided sessions indicate substantial process-related reflections as well as a sharing and synthesizing of the collected results, experiences and insights with colleagues. What becomes observable here are parts of the linked mutually reinforcing double loop learning processes from Section 2 associated with the two challenges that universities face in managing the digital transformation. Applying this theory further and broadening the perspective, additional learnings can be derived, which may be observable in the mid-term, as illustrated in Figure 7:

The formulated strategic objective of providing opportunities for campus citizens to gather practical know-how in digital transformation, in order to, as a university, better cover this cross-sectional topic in curriculum and research (“what”), led to the measure of conducting a PoC, where selected campus citizens act as citizen developers. The considered campus citizen brought in a need arising from his teaching activities. This need has been addressed by the measure to have him develop a small utility app (“how”), as described. The experiences he gathered then implied some learnings, first regarding his teaching process, but also regarding the topic of digital transformation in general. Ideas on the further development of process and app now might lead to further measures. Moreover, he might use his new insights on digital transformation also in his other courses, and further share them - e.g. with some academic program management representatives at an internal convention -, which would then, if perhaps only slightly, impact the university’s common understanding of what the digital transformation is about, and how digitalization could be further used strategically for teaching. This would finally establish the various learning loops from Figure 1.

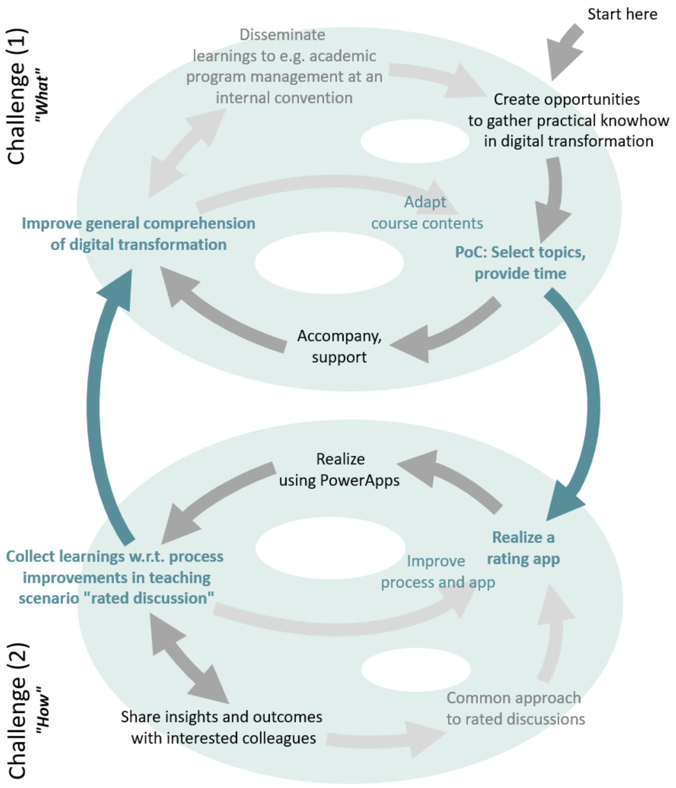


Figure 7: Mapping the described exemplary outcome of the PoC to the linked double loop learning processes

5. CONCLUSIONS AND OUTLOOK

A holistic architecture for a digital campus is described comprising all building blocks necessary to digitally transform a university while addressing two challenges: The transversal topic of digitalization needs to be covered well-balanced in teaching as well as research, and the university must find a way to exploit its own digitalization potential in the best possible way.

In tackling both challenges, mutually reinforcing effects can be exploited by involving academic staff and students in a practical way to contribute to the university’s digital transformation. As one possible option, these campus citizens are motivated to act as citizen developers. For their development work, appropriately prepared and supported low-code development environments are provided. The digital solutions developed by them are applied within the university itself - first by a smaller audience, in an experimental way, later, when they have proven and stabilized, possibly by a larger part of the university. Thus, citizen developers help to scale up the digital transformation of the university. A proof of this concept is underway, the illustrated first result gives promising indications of feasibility and coherence. Further analyses will help to better understand the requirements for a laboratory-like environment in which a fruitful citizen development can take place. Citizen development is also to be integrated into course work; initial experiments were very well received by the students.

In parallel, further options for the practical participation of campus citizens in the digital transformation of the university will be discussed and implemented. Once a sufficient number of citizen-developed solutions and other practical contributions to the university’s digital transformation process are available, empirical studies are planned to measure the impact of the active “digital” involvement of campus citizens within their university on the knowledge conveyed by the university about the phenomenon of digital transformation and, with second priority, to what extend this topic is given adjusted consideration in research.

Numerous discussions have accompanied the authors in the preparation of the presented thoughts, for example with Andreas Müller, Tom Kleiber, Lothar Ritter, Stefan Richter, Roman Hänggi, Reto Eugster and Peter Jaeschke. Many thanks for that.

6. REFERENCES

Argyris, C. (1991). Teaching Smart People How to Learn. *Harvard Business Review*.

Beynon-Davies, P., Carne, C., Mackay, H., & Tudhope, D. (1999). Rapid application development (RAD): an empirical review. *European Journal of Information Systems, 8*(3).

Bonk, C., & Graham, C. (2012). *The Handbook of Blended Learning: Global Perspectives, Local Designs*. John Wiley & Sons.

Bredlow, C. (2019). *Bei der Digitalisierung ist es wie bei der Musik*. Retrieved from https://digitalmindset.de/inspiration/keynotes

Cross, N. (2011). *Design thinking: Understanding how designers think and work*. Oxford: Berg.

Deuze, M. (2006). Participation, Remediation, Bricolage: Considering Principal Components of a Digital Culture. *The Information Society, 22*(2).

Everhard, J. (2019). *The Pros And Cons Of Citizen Development*. Retrieved from www.forbes.com

Fielding, R. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. Retrieved from https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation_2up.pdf

Hemerling, J., Kilmann, J., Danoesastro, M., Stutts, L., & Ahern, C. (2018). *It's Not a Digital Transformation Without a Digital Culture*. Retrieved from https://www.bcg.com/en-us/publications/2018/not-digital-transformation-without-digital-culture.aspx

Iijima, K., & others. (2019). *Magic Quadrant for Enterprise Low-Code Application*. Gartner.

Johnson, W. (2017). *A Business Without Millennials is a Business Without a Future*. Von https://www.linkedin.com/pulse/business-without-millennials-future-whitney-johnson/ abgerufen

Licka, P., & Gautschi, P. (2017). *Survey: The digital future of higher education – What does it look like and how can it be shaped?* Zürich, Köln: Berinfor.

OECD. (2017). *OECD Science, Technology and Industry Scoreboard 2017: The digital transformation*. Paris: OECD Publishing. Retrieved from http://dx.doi.org/10.1787/9789264268821-en

Ost - Eastern Switzerland University of Applied Sciences. (2020). Retrieved from https://www.ost.ch/en/

Papert, S. (1993). *The children's machine: rethinking school in the age of the computer*. New York: Basic Books.

Parecki, A. (2012). *OAuth 2.0*. Retrieved from https://oauth.net/2/

Richardson, C., & Rymer, J. (2014). *New Development Platforms Emerge For Customer-Facing Applications*. Forrester.

Richardson, C., & Rymer, J. (2016). *The Fractured, Fertile Terrain Of Low-Code Appl. Platf*. Forrester.

Sheridan, K., Rosenfeld Halverson, E., Litts, B., Brahms, L., Jacobs-Priebe, L., & Owens, T. (2014). Learning in the making: A comparative case study of three makerspaces. *Harvard Educational Review, 84*(4).

Stone, S. (2019). *Digitally Deaf - Why Organizations Struggle with Digital Transformation*. Springer.

Swagger. (2016). *About Swagger Specification: What is OpenAPI?* Retrieved from https://swagger.io/docs/specification/about/

Tucker, B. (2012). The flipped classroom. *Education Next*.

Völter, M., Stahl, T., Bettin, J., Haase, A., & Helsen, S. (2013). *Model-Driven Software Development: Technology, Engineering, Management*. John Wiley & Sons.

7. AUTHORS' BIOGRAPHIES

Christoph Baumgarten studied computer science at the Technical University of Braunschweig, Germany, at master's level (1995), and gained research experience as a visiting scholar at the Arizona State University as well as ETH in Zurich, Switzerland. In 1999 he received a PhD at the Technical University of Dresden, Germany, within a state-funded Research Training Group. His professional experience comprises various IT management positions, including CIO of the University of St. Gallen, Switzerland (2009-2017), and Head Capability Development AIM (Aeronautical Information Management) at Skyguide Swiss Air Navigation Services (2003-2009). Since mid-2017 Christoph Baumgarten is working as a lecturer in the economics department of the FHS St. Gallen.

Alex Simeon studied mechanical engineering at the HSR in Rapperswil, Switzerland, from 1982-85. He then worked as an assistant and from 1987-90 as a systems engineer in the mechanical engineering department. In 1991 he moved to Sulzer Ltd., where he held management positions in various departments and was responsible for the introduction of a wide range of IT systems. From 2000-2003 he was head of CAx/PDM systems at Sulzer's IT department. Alex Simeon was elected Professor of Mechanical Engineering and Head of IPEK (Institute for Product Design, Development and Construction) at HSR in 2003. Since September 2011, he has been Vice-Rector for Applied Research and Development and a member of the HSR's university management. In September 2020 he will take up his new position as Chief of Staff of the OST – Eastern Switzerland University of Applied Sciences.

Michael C. Wilhelm studied mechanical engineering at the Technical University of Karlsruhe, Germany, today KIT. From 1983 to 1988 he worked at the Institute of Production Science. His work focused on IT-Systems and methods for development and production, with focus on CAx-Systems for design and simulation, CAM (manufacturing) and CAQ (Quality)-Systems. After completing his doctorate, he worked as a consultant for automotive and suppliers as well as for mechanical engineering companies. In 1997 he moved to the University of Applied Sciences in Karlsruhe where he give lectures for Quality Management and Production. Since 2015 he is member of the university management and responsible for teaching at the Interstate University NTB in Buchs, Switzerland.