



UMEÅ UNIVERSITY

Architecture Is Not Enough

Bridging the Gap Between Design and Real Work

Karin Brodén, PhD

karin.broden@umu.se

Informatics, Umeå University

Karin Brodén, PhD



Researcher in Informatics, Umeå University

Long background in the IT industry developing systems critical to society. Now researching what happens when systems meet everyday work.

RESEARCH & TEACHING



INDUSTRY



Today I will argue for three things

- 1. Systems do not fail only because of poor architecture**
They can fail because real work is more complex than the assumptions built into the architecture.
- 2. Workarounds, informal practices and translation gaps are not noise**
They are evidence of how the system actually functions in practice.
- 3. Architecture should be judged by the work it enables**
Not only by technical coherence, but by how well it supports people, tasks and organisational reality.

From architecture on paper to architecture in use.

A system can work exactly as designed — and still not work



Technically correct.



Organisationally difficult.



Practically adapted.

Debatt

”Sätt medarbetarna i fokus när digitala verktyg införs”

Medarbetare inom kommuner är motiverade att föra digitaliseringen framåt, men får inte det stöd som de behöver för att göra det, skriver forskaren Karin Brodén.

”ttest om människor” - Dagens Samhälle

mhälle

& beslut Samhälle & välfärd Offentlig ekonomi Chef & arbetsgivare Opi

Debatt

”Vi måste våga prata om digitaliseringens vardag”

Ska offentlig sektor lyckas fullt ut med sina digitala ambitioner krävs en skarp omsvängning från vision till verkstad, skriver forskaren Karin Brodén.

Publicerad: 21 augusti 2025, 05:30

Det här är opinionsmaterial

Åsikterna som uttrycks här står skribenten/skribenterna för.





What happens when architecture meets work?

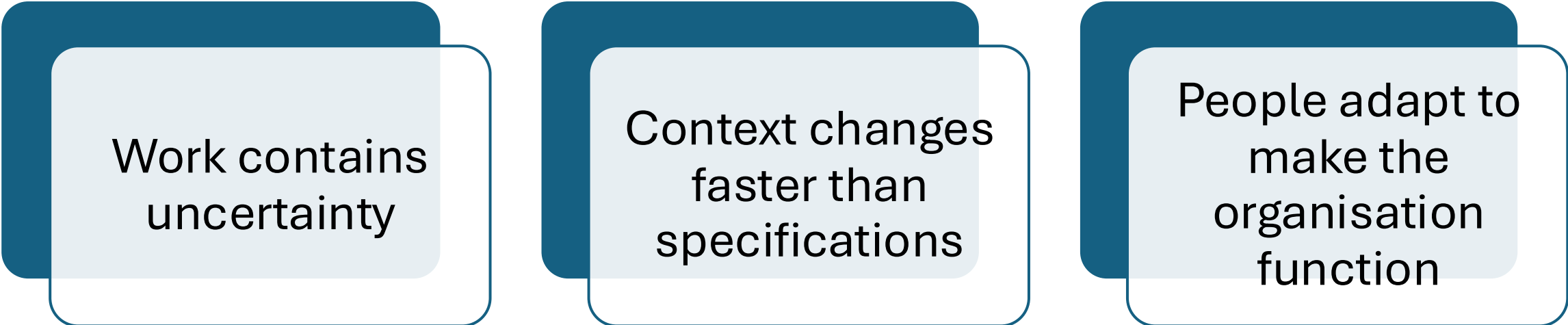
System as designed

- Processes are defined
- Roles are clear
- Data flows are known
- Interfaces are specified
- Exceptions are manageable

System as used

- Work is interrupted
- Roles overlap
- Information travels informally
- Local practices emerge
- Exceptions become everyday work

The gap is not an implementation failure



Work contains
uncertainty

The diagram consists of three identical rectangular boxes arranged horizontally. Each box has a dark blue header and a light blue body. The text is centered within each box. The first box contains 'Work contains uncertainty', the second 'Context changes faster than specifications', and the third 'People adapt to make the organisation function'.

Context changes
faster than
specifications

People adapt to
make the
organisation
function

Four things I repeatedly see in digitalisation projects



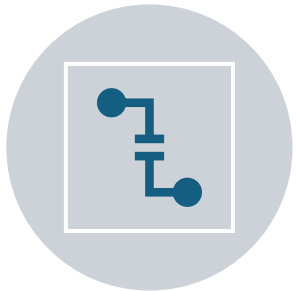
Informal change

Formal implementation is only part of the change process.



Translation gaps

Someone must translate between technology, work and organisational needs.



Integration & dependency

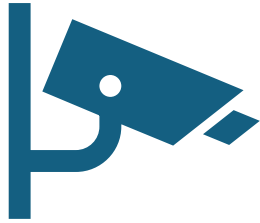
A good application can fail inside a fragmented system landscape.



Workarounds

People create alternative practices to get the work done.

Case: when the physical world does not fit the system



Designed: A sensor provides information about what is happening in the apartment.



Reality: Room layout limits the field of view.



Response: Staff create a workaround so they can still interpret what is happening.

Is this a user problem, a technology problem — or an architecture problem?

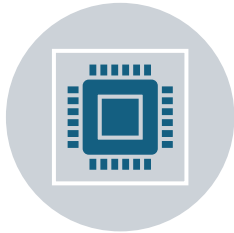
A workaround is data



Workarounds show where the designed system and the working system diverge.



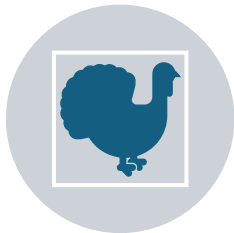
What does the workaround compensate for?



What knowledge does the user have that the system does not?



What risk does the workaround introduce?



Should it be removed — or learned from?

There are two architectures



Formal architecture

Systems, interfaces, components, standards, processes



Operational architecture

People, routines, spreadsheets, messages, tacit knowledge, exceptions



Organisations run on both.

When people struggle with a system, training may not be the answer

Knowledge problem
“I do not know how.”

Design problem
“The system does not support the task.”

Organisational problem
“The rules, responsibilities or conditions make this difficult.”

These require very different interventions.

Case: patient data in healthcare

The study

- Healthcare staff in two clinics: women's health care, and anaesthesia, surgery & intensive care
- Self-assessment survey (~130 responses) combined with think-aloud tests
- Focus: how digital support for patient data is experienced in practice — not how the systems are meant to work

Brodén, Ängeby & Nordin · Umeå University & Karlstad University

Stable in everyday work.

Vulnerable when it gets complex.

Uncertainty arises around:

- Protected identities
- Requests from authorities
- Transferring data between systems
- Situations where system support and routines are missing

Everyone wants to do it right.

The missing role: translation



Technology ←
Translation
capability → **Work**



Understand
technical
possibilities



Understand actual
work



Understand
organisational
consequences



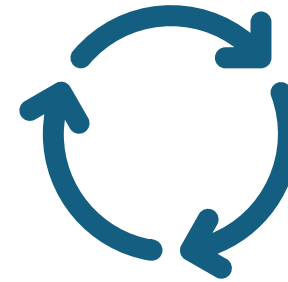
Recognise when
requirements have
changed

Implementation is not an event



Project view

Design → Build → Go-live. Done



Practice view

Go-live → Adaptation → New practices →
New requirements → Redesign

Integration is experienced at the task level

An architecture diagram may say systems are integrated.
A worker may still have to:

Re-enter information

Search across
several systems

Remember which
system contains
what

Resolve inconsistent
information

Contact another
person to complete
the task



Technical integration and experienced integration are not the same thing.

Architecture decisions can redistribute work

- Every architectural decision creates work somewhere.
- Examples:
- More standardisation → less local flexibility
- More security controls → more authentication work
- More data quality requirements → more registration work
- Fewer integrations → more manual coordination
- More automation → more exception handling

Questions architects can ask differently



What work will this architecture create?



Where will exceptions occur?



Who will resolve them?



What knowledge must remain outside the system?



Which informal practices are currently holding the process together?



What happens when the system meets time pressure?



How will we learn after go-alive?

From system architecture to working architecture



From systems as specified
To systems in use



From implementation
To continuous adaptation



From user compliance
To understanding work



Good architecture is not only structurally sound. It remains viable when it encounters reality.

Architecture creates possibilities. Work reveals reality.



The question is not only:
“Is this system well designed?”



It is also:
“What happens when people actually have to work with it?”

Questions & contact



karin.broden@umu.se



I welcome collaboration
with organisations, for
example as:



Research partner in
practice-based projects



Embedded researcher
(ongoing/formative
evaluation)



Seminars and
workshops