

# Evolutionary Architecture and Architectural Governance

**Rebecca Parsons**

Chief technology officer



We're actually pretty good  
at requirements change  
**but...**

 /thoughtworks

@rebeccaparsons

# What about ecosystem change?



@rebeccaparsons

**How is long-term  
planning possible  
under constant change?**

# What is evolutionary architecture?

**/thoughtworks**

@rebeccaparsons

**An evolutionary architecture  
supports guided incremental  
change across multiple  
dimensions**

**/thoughtworks**

@rebeccaparsons

An evolutionary architecture  
supports **guided** incremental  
change across multiple  
dimensions

/thoughtworks

@rebeccaparsons

**An evolutionary computing fitness function characterizes how close a solution is to the desired result**

**/thoughtworks**

@rebeccaparsons



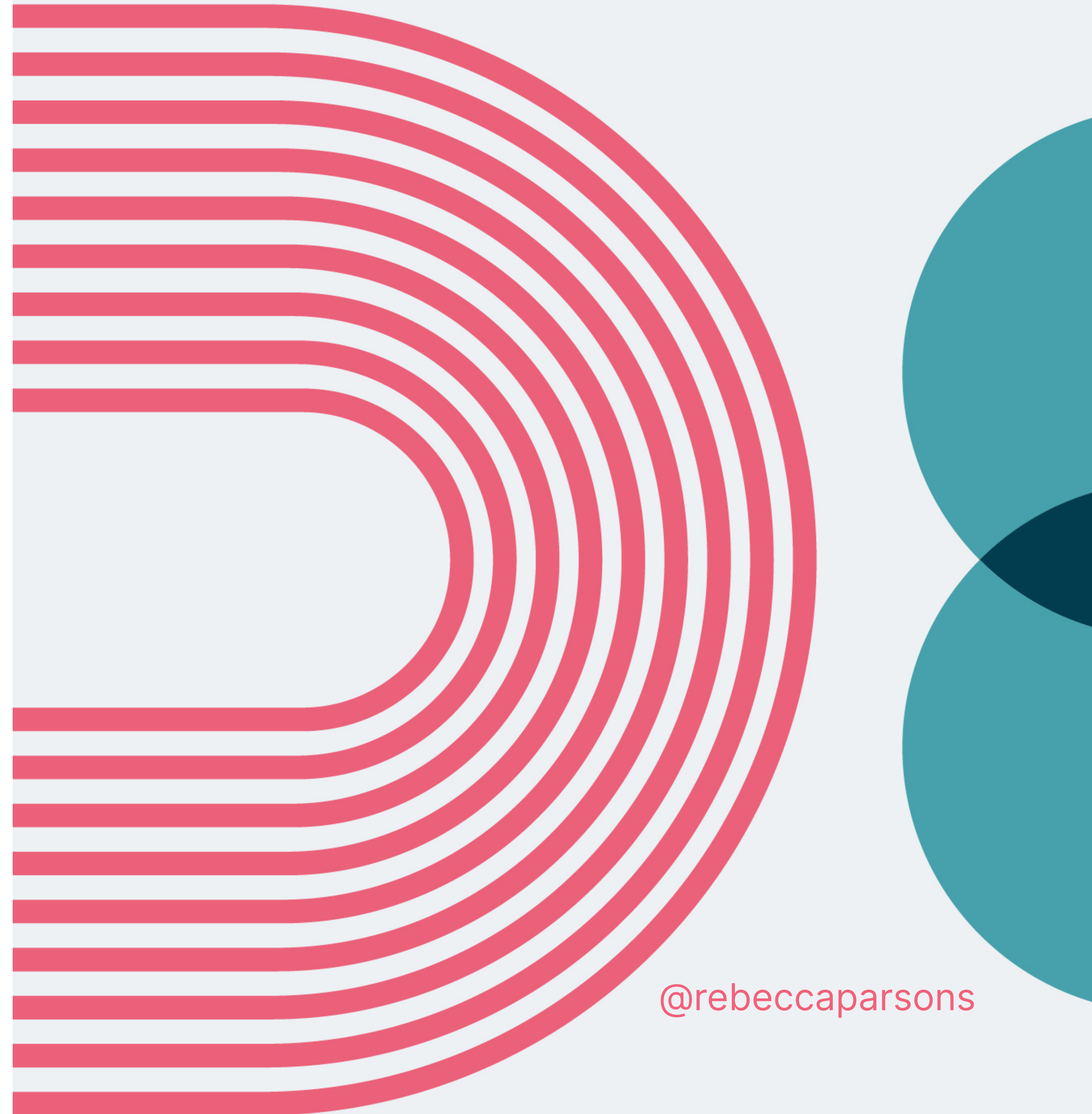
**An architectural fitness  
function characterizes  
how close a system is to  
the desired architectural  
characteristics**

**/thoughtworks**

@rebeccaparsons

# Example fitness functions

- Cyclic dependencies
- Consumer driven contracts
- Caching with staleness
- Monitoring
- Synthetic transactions
- Chaos monkey



@rebeccaparsons

An evolutionary architecture  
supports guided **incremental**  
**change** across multiple  
dimensions

/thoughtworks

@rebeccaparsons

An evolutionary architecture  
supports guided incremental  
change across **multiple**  
**dimensions**

/thoughtworks

@rebeccaparsons

# -ilities

accessibility  
accountability  
accuracy  
adaptability  
administrability  
affordability  
agility  
auditability  
autonomy  
availability  
compatibility  
composability  
configurability  
correctness  
credibility  
customizability  
debugability

degradability  
determinability  
demonstrability  
dependability  
deployability  
discoverability  
distributability  
durability  
effectiveness  
efficiency  
reliability  
extensibility  
failure  
transparency  
fault-tolerance  
fidelity  
flexibility

inspectability  
installability  
integrity  
interchangeability  
interoperability  
learnability  
maintainability  
manageability  
mobility  
modifiability  
modularity  
operability  
orthogonality  
portability  
precision  
predictability

process  
capabilities  
producibility  
provability  
recoverability  
relevance  
repeatability  
reproducibility  
resilience  
responsiveness  
reusability  
robustness  
safety  
scalability  
seamlessness  
self-sustainability  
serviceability

supportability  
securability  
simplicity  
stability  
standards  
compliance  
survivability  
sustainability  
tailorability  
testability  
timeliness  
traceability  
transparency  
ubiquity  
understandability  
upgradability  
usability

# -ilities

accessibility  
accountability  
accuracy  
adaptability  
administrability  
affordability  
agility  
auditability  
autonomy  
availability  
compatibility  
composability  
configurability  
correctness  
credibility  
customizability  
debugability

degradability  
determinability  
demonstrability  
dependability  
deployability  
discoverability  
distributability  
durability  
effectiveness  
efficiency  
reliability  
extensibility  
failure  
transparency  
fault-tolerance  
fidelity  
flexibility

inspectability  
installability  
integrity  
interchangeability  
interoperability  
learnability  
maintainability  
manageability  
mobility  
modifiability  
modularity  
operability  
orthogonality  
portability  
precision  
predictability

## evolvability

process  
capabilities  
producibility  
provability  
recoverability  
relevance  
repeatability  
reproducibility  
resilience  
responsiveness  
reusability  
robustness  
safety  
scalability  
seamlessness  
self-sustainability  
serviceability

supportability  
securability  
simplicity  
stability  
standards  
compliance  
survivability  
sustainability  
tailorability  
testability  
timeliness  
traceability  
transparency  
ubiquity  
understandability  
upgradability  
usability

# Principles

**/thoughtworks**

@rebeccaparsons

# Last responsible moment

**/thoughtworks**

@rebeccaparsons



# Architect and develop for evolvability

**/thoughtworks**

@rebeccaparsons

# Postel's Law

**/thoughtworks**

@rebeccaparsons

# Architect for testability

**/thoughtworks**

@rebeccaparsons

# Conway's Law

**/thoughtworks**

@rebeccaparsons

# Techniques

**/thoughtworks**

@rebeccaparsons

# Database refactoring

**/thoughtworks**

@rebeccaparsons

# Choreography

**/thoughtworks**

@rebeccaparsons

# Contract testing

**/thoughtworks**

@rebeccaparsons



# Continuous delivery

**/thoughtworks**

@rebeccaparsons

# Evolvability and Experimentation

**/thoughtworks**

@rebeccaparsons

# Platform for Experimentation

**/thoughtworks**

@rebeccaparsons

# Culture of Experimentation

**/thoughtworks**

@rebeccaparsons

# Evolvability of different software architectures

**/thoughtworks**

@rebeccaparsons

# Big ball of mud

**/thoughtworks**

@rebeccaparsons

# Microservices

**/thoughtworks**

@rebeccaparsons

# Structured monolith

**/thoughtworks**

@rebeccaparsons



# Layered monolith

**/thoughtworks**

@rebeccaparsons

# Micro-kernel

**/thoughtworks**

@rebeccaparsons

# Governance



@rebeccaparsons

# Fitness function driven



@rebeccaparsons

# Outcomes not implementations

**/thoughtworks**

@rebeccaparsons

# Mechanics



@rebeccaparsons

# Mechanics

- Define your architectural fitness function
- Select a dimension you're most worried about
- Start improving on that dimension
- Focus on what matters most
- Monitor trends, adapt and repeat



# You need to evolve your fitness functions too



@rebeccaparsons



# Thank you

**Rebecca Parsons**

*@rebeccaparsons*

thoughtworks.com

The logo for Thoughtworks, featuring a red diagonal slash followed by the word "thoughtworks" in a white, lowercase, sans-serif font.