

# **Modernizing legacy teams**

**Andreas Möller**

**Andreas Möller**  
**@localheinz**

# Agenda

1. Introduction
2. Legacy applications
3. Legacy teams
4. Conclusion

**Identifying  
legacy  
applications**

**“Legacy code is  
code without tests”**

Michael Feathers

**"33% of developers  
do not write tests  
for PHP"**

[jetbrains.com/lp/devecosystem-  
2022/php/](https://jetbrains.com/lp/devecosystem-2022/php/)

# Diagnosis

# Version control



# Dependency manager

# Dependencies

**PHP version**

# PHP extensions

# **Packages and frameworks**

# Development tools

# Runner for development tools

# **Configuration for Continuous Integration system**



# Monitoring and logging

**PHP code**

**Objective vs.  
subjective criteria**

**Criteria change  
over time**

**Criteria change  
with experience**

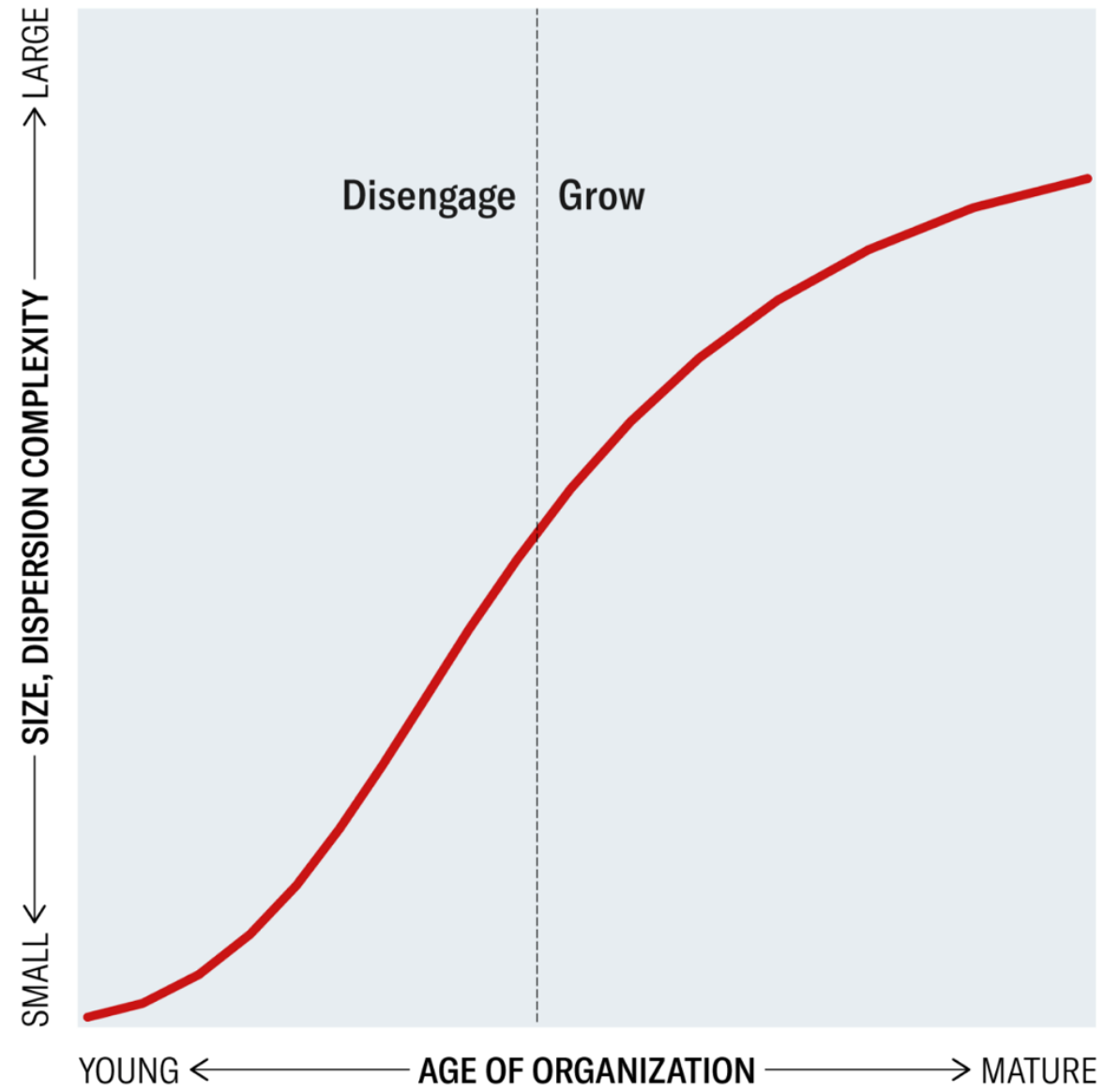
**Normalizing  
legacy  
applications**

# Five stages of growth for small businesses

[hbr.org/1983/05/the-five-  
stages-of-small-business-growth](https://hbr.org/1983/05/the-five-stages-of-small-business-growth)

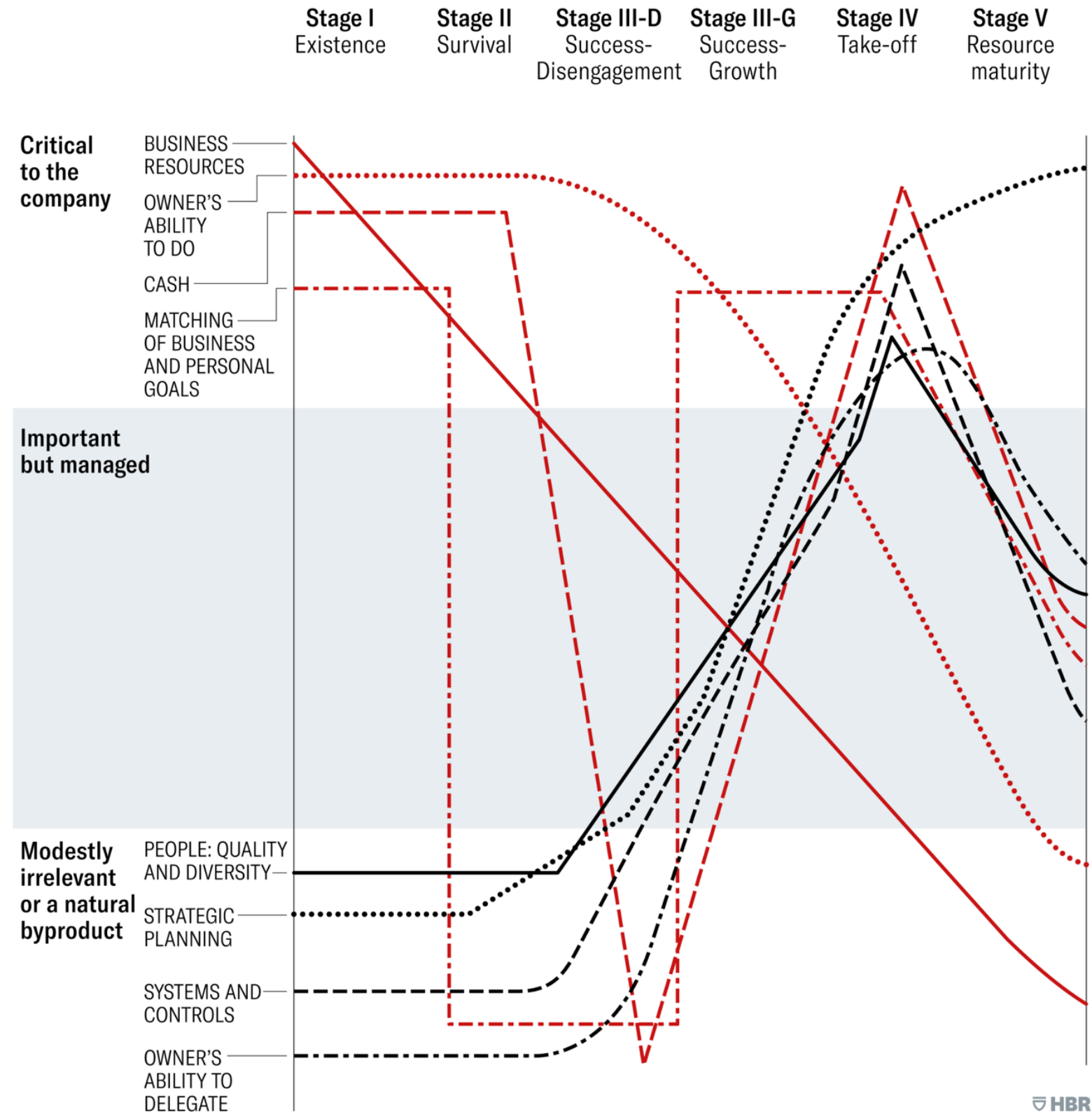
# Exhibit 2: Growth Stages

Stage I Existence    Stage II Survival    Stage III Success    Stage IV Take-off    Stage V Resource maturity





# Exhibit 5: Management Factors and the Stages



**Mismatch**

**Avoiding  
legacy  
applications**

**Career change**

**Job hopper**

**Accept and  
embrace**

# **Modernizing legacy applications**

**"No matter what  
they tell you, it's  
always a people  
problem"**

Jerry Weinberg



**Treating effects**

**Treating causes**

# Identifying legacy teams

# Exhibit A: Apprentice

Customer Support

# Exhibit B: Junior Developer

Mob Programming

# Exhibit C: Developer

Carnival Cash

# Exhibit D: Senior Developer

Git Oops

# Exhibit E: Team lead

9 to 5



# Exhibit F: Architect

Class-based Programmer

# **Exhibit G: Engineering Manager**

Expert Beginner

# Exhibit H: Engineering Manager

Hacker

# Exhibit F: Owner

Analysis Paralysis

# 10x programmer

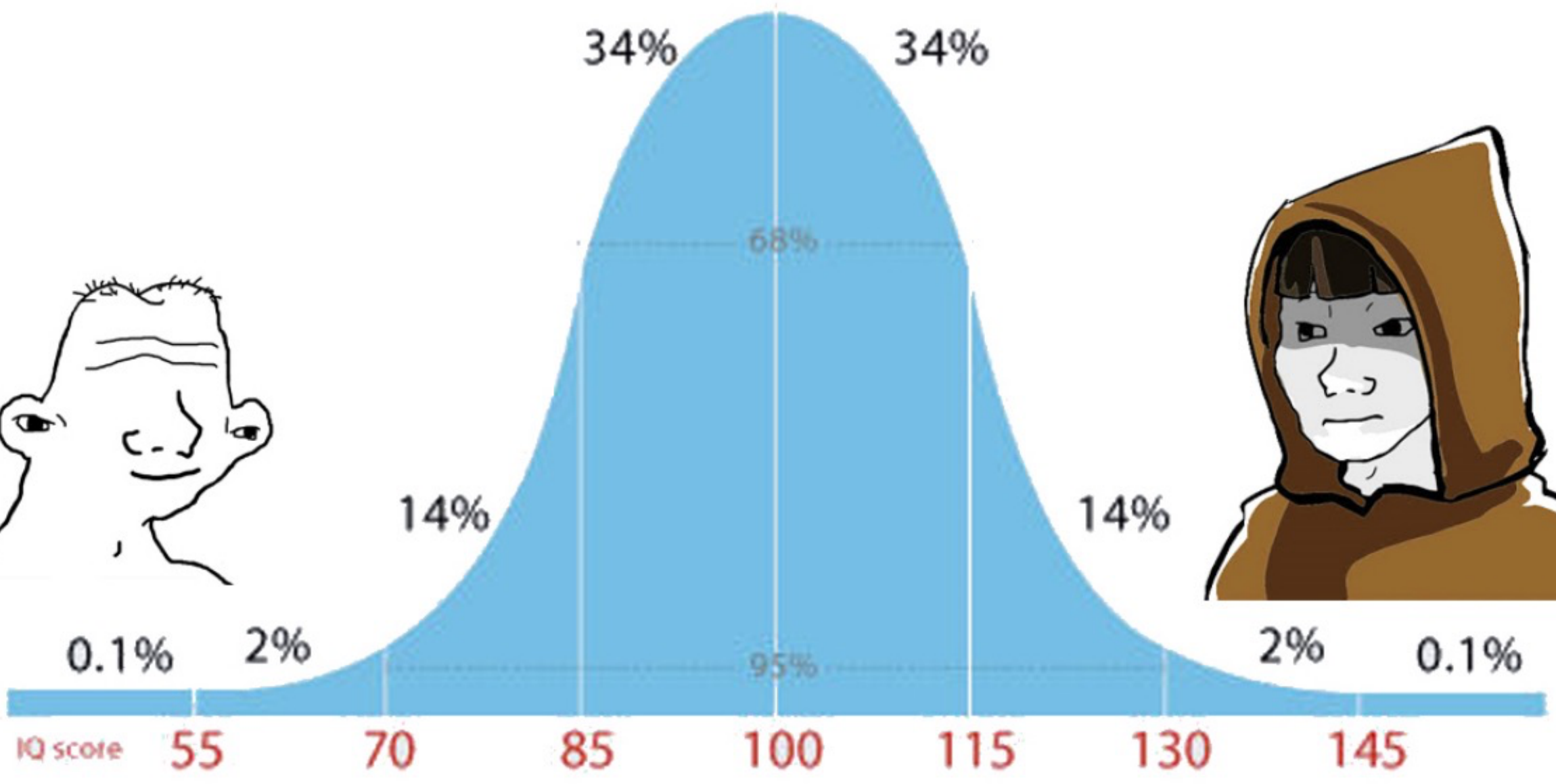
Tom De Marco & Timothy Lister

# 0.1x programmer

Tom De Marco & Timothy Listener

# 10x organization

Tom De Marco & Timothy Listener





**Modernizing  
legacy teams**

**Getting help**

**Acknowledging  
problems**

**Scaling down**

# Conclusion

1. Software rots
2. Modernizing legacy applications is hard
3. Modernizing legacy teams is hard
4. Software is hard
5. Context matters

# Thank you!

[am@localheinz.com](mailto:am@localheinz.com)

[twitter.com/@localheinz](https://twitter.com/@localheinz)

[github.com/localheinz](https://github.com/localheinz)

[phpc.social/@localheinz](https://phpc.social/@localheinz)

[localheinz.com](https://localheinz.com)