



Intro to Developer Productivity Engineering (DPE)



Jesse Gorzinski

Senior Business Architect

Power Systems Development

jgorzins@us.ibm.com | [@IBMJesseG](https://twitter.com/IBMJesseG)



Justin Reock

Chief Evangelist and Field CTO
Gradle Enterprise

Justin has over 20 years of experience working in various software roles including JEE work. He is an outspoken free software evangelist, delivering enterprise solutions, technical leadership, various publications and community education on databases, architectures, and integration projects.



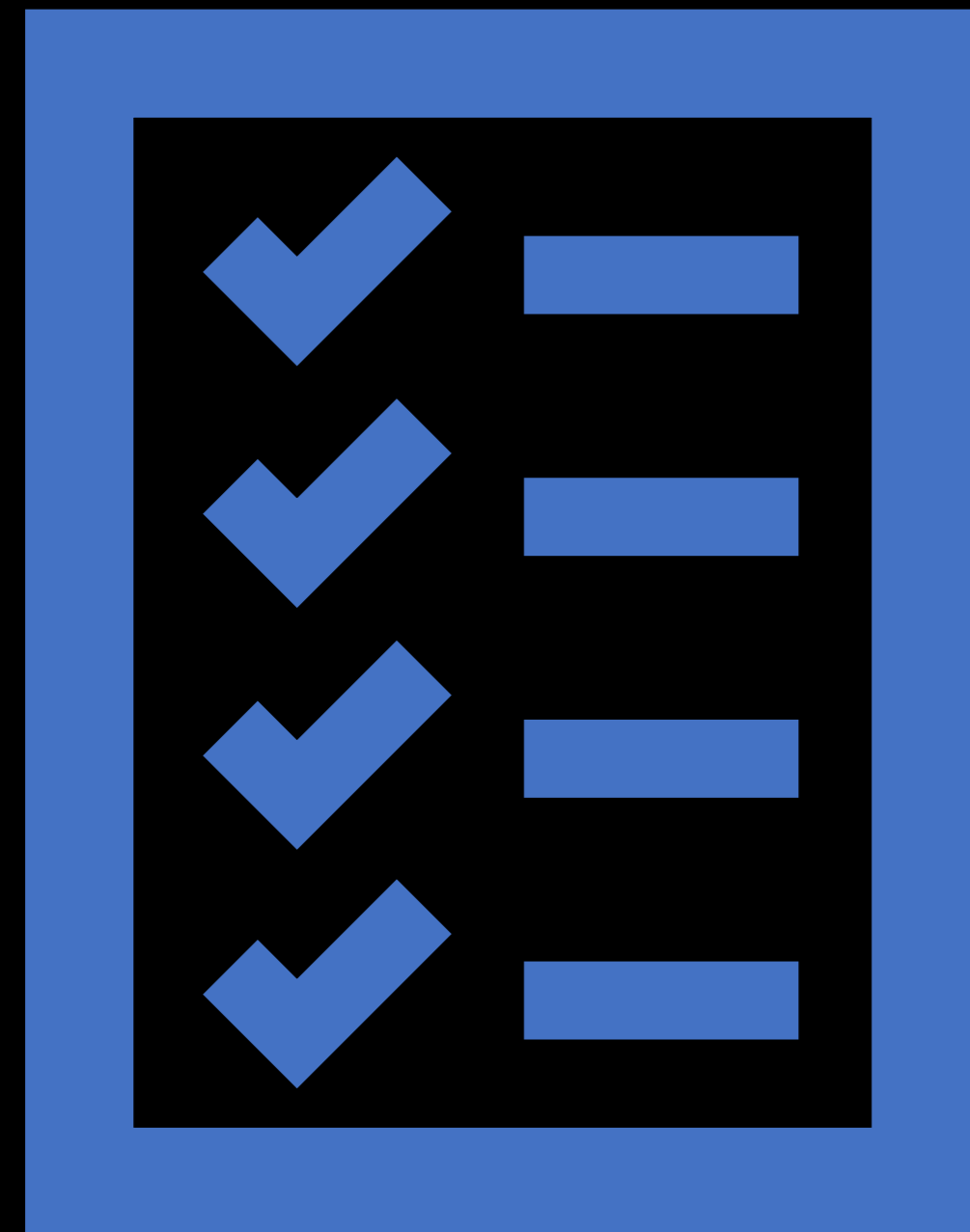
[@justinreock](#)

Digital Transformation (DX) still a thing



29 Oct 2020

IDC Reveals 2021 Worldwide Digital Transformation Predictions; 65% of Global GDP Digitalized by 2022, Driving Over \$6.8 Trillion of Direct DX Investments from 2020 to 2023



Agenda

- What is DPE?
- Measuring productivity
- Building 10x developers
- What can you do?

What is Developer Productivity Engineering?

"Developer Productivity Engineering (DPE) is a software development practice used by leading software development organizations to maximize developer productivity and happiness."

(from the Developer Productivity Engineering handbook
<https://gradle.com/developer-productivity-engineering/>)

1

Productivity

2

Happiness



**“It’s no longer the big beating the small,
but the fast beating the slow.”**

Eric Pearson, CIO, InterContinental Hotels Group





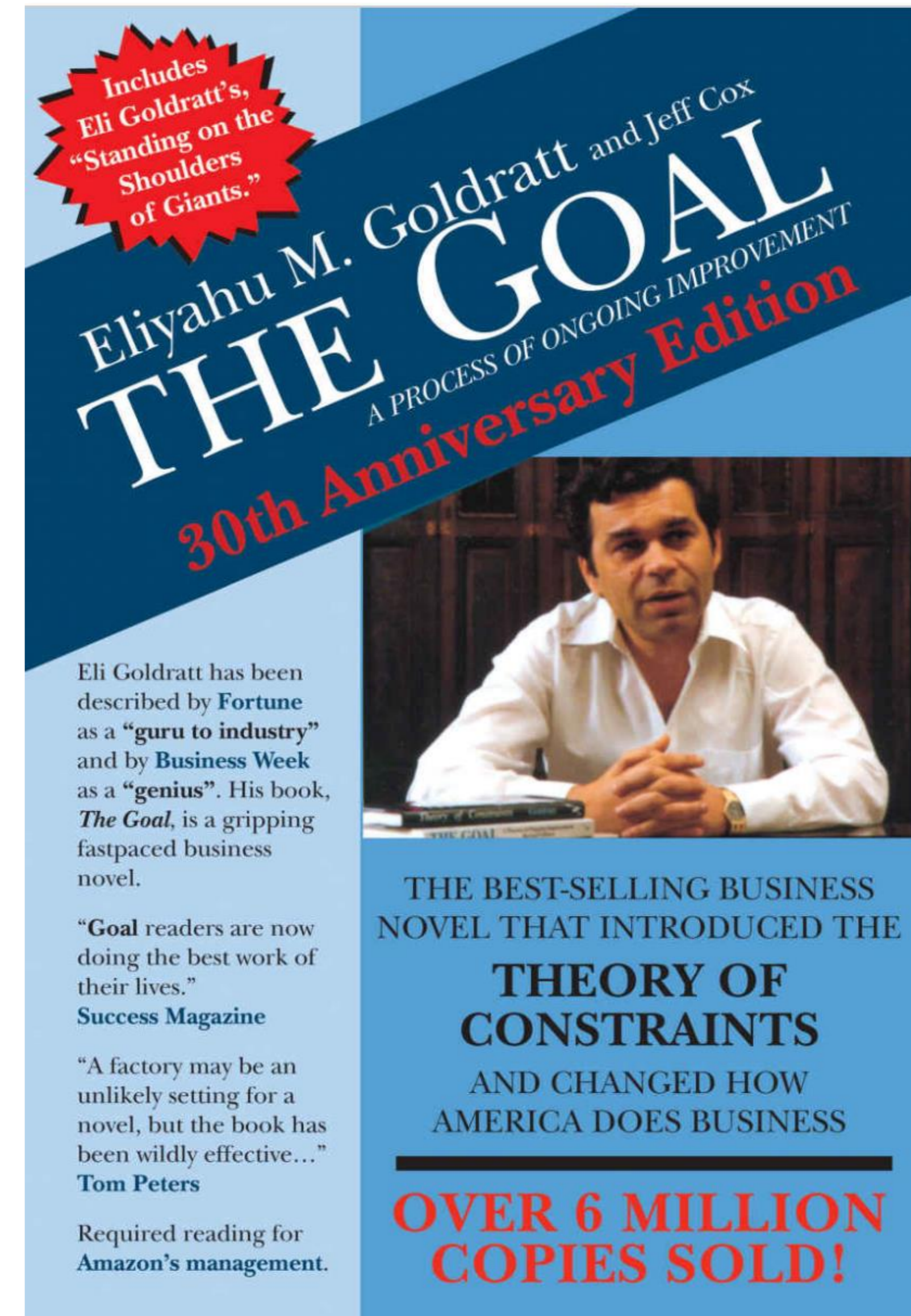
The Ancient Business Wisdom

...Of the 70s and 80s

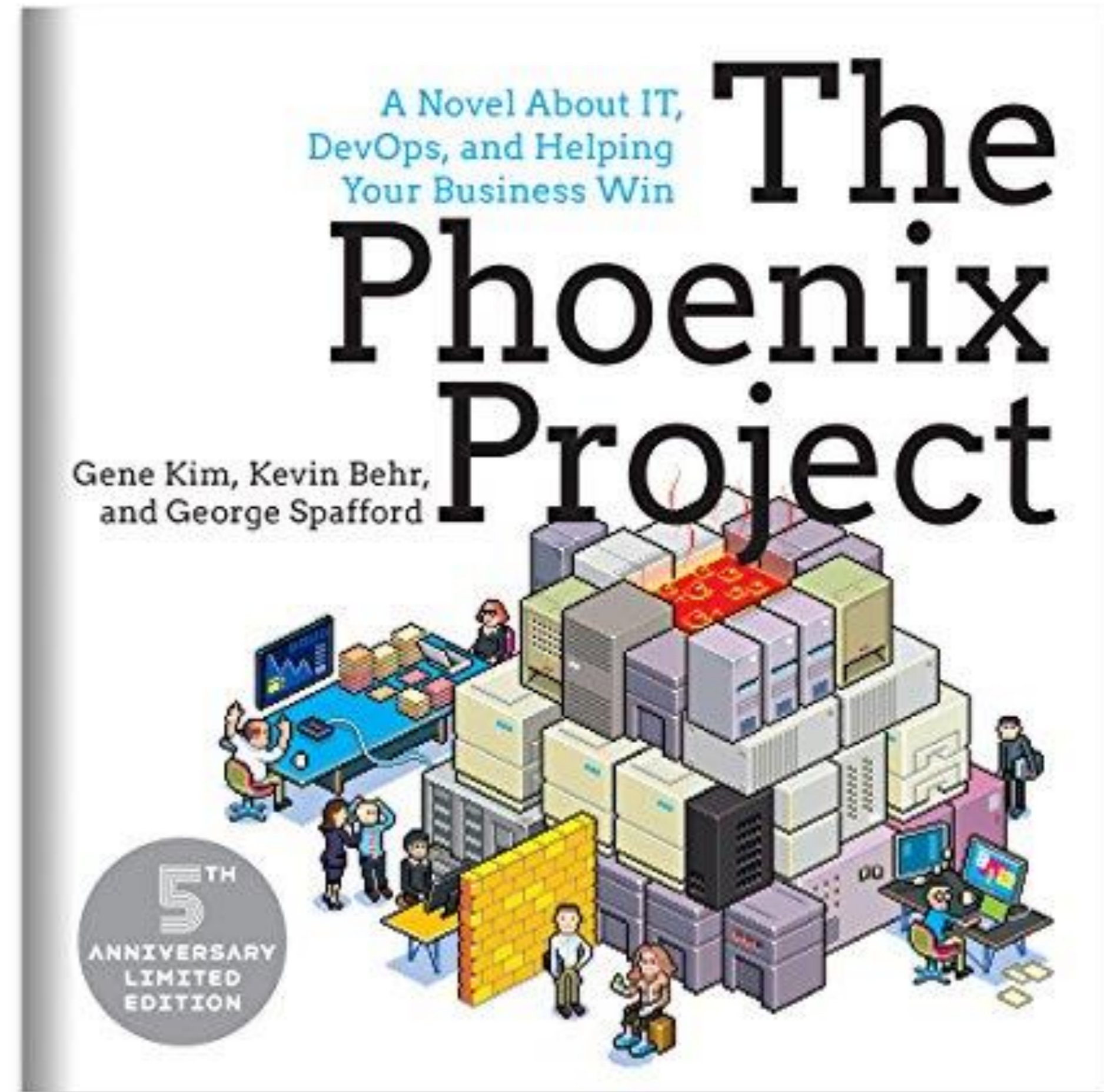


The Goal

- Dr. Goldratt was a **physicist turned business novelist**
- His most famous work is the **Theory of Constraints**, which has helped form the foundations of modern business productivity theory
- **DevOps, Agile, Lean, etc**, all stem from the Theory of Constraints
- The theory focuses on treating organizations as complex machines, and **eliminating bottlenecks** to improve organizational efficiency



- **Modernizes the Theory of Constraints** for software organizations
- A fun read, business drama, **not textbook** format
- Demonstrates the relationship between **DevOps** and **throughput optimization**
- **Teaches the importance** of VSM, observability, and continuous improvement



1970s+

JIT
Manufacturing

1980s+

Business
Process
Reengineering

1990s+

Change
management

2000s+

Agile, Lean Six
Sigma

2010s+

DevOps

2020+

DPE

DPM: a competing framework

"Developer Productivity Management (DPM) focuses on the people, and answers questions like, 'How can we get more output out of individual developers and teams by defining and tracking the right metrics? Such metrics typically help to quantify output, evaluate performance and competencies, build and manage teams, and optimize collaboration and time management.

(from the Developer Productivity Engineering handbook
<https://gradle.com/developer-productivity-engineering/>)



Management

Low productivity



Pizza Party!



Developer Productivity Engineering Focus



STANLEY

18

m&m's

INTERSTATE BATTERIES

Rheem

AMERICAN ETHANOL

MONSTER ENERGY
MECHANIX WEAR
3M
K&N
LINCOLN WELDERS

MOOG
Edelbrock
SUNOCO

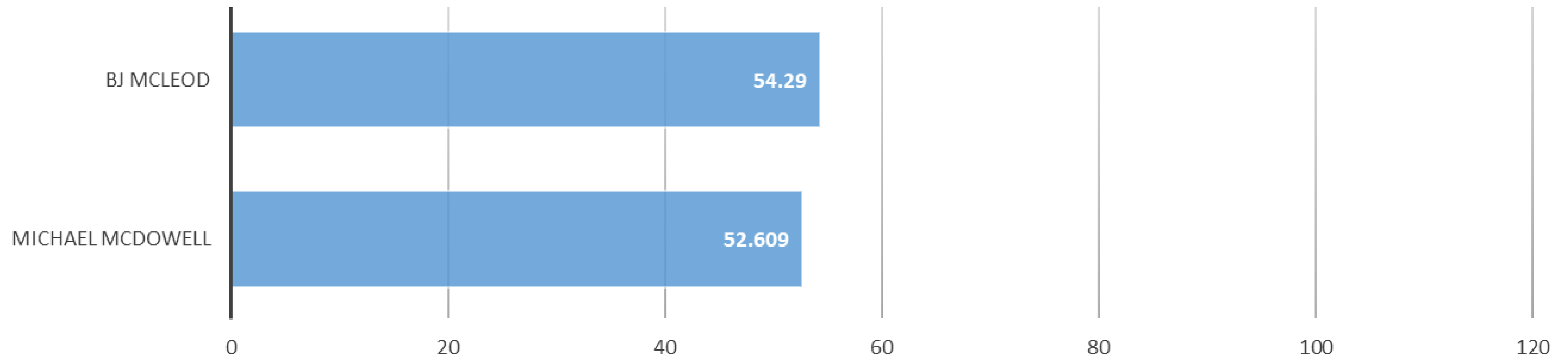
GOODYEAR

Mobil 1
FREIGHTLINER
SIEMENS
DOOSAN
LINCOLN WELDERS
MAC TOOLS
Cessna
SHERWIN WILLIAMS

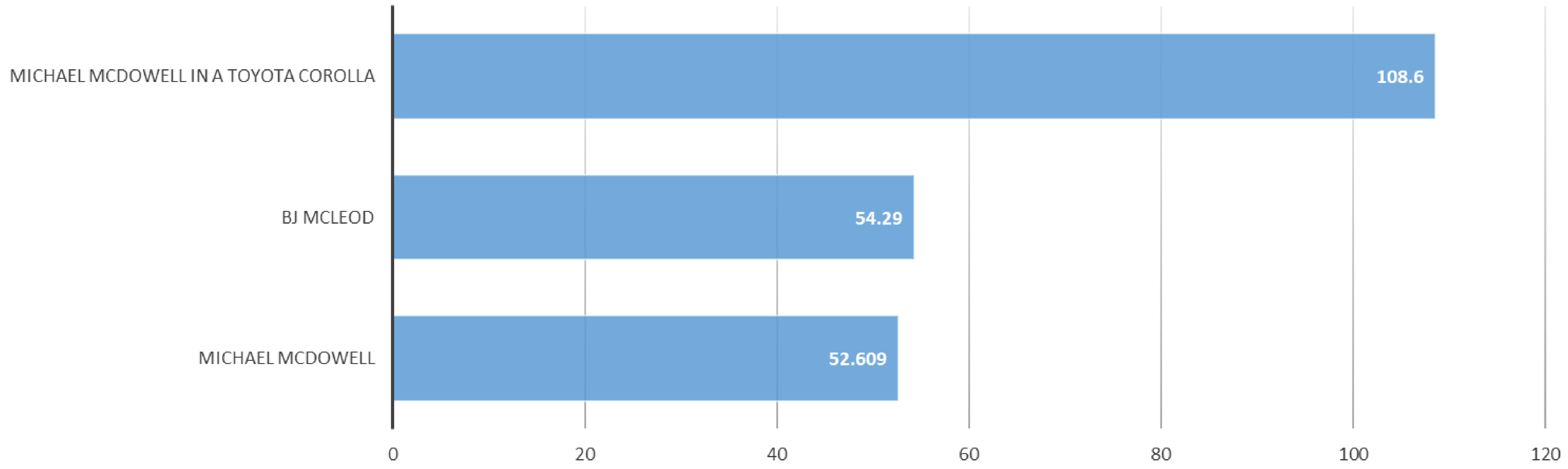
NASCAR RACE CAR

TOYOTA
SUNOCO
SUNOCO
SUNOCO

Lap times at Talladega (s)



Lap times at Talladega (s)



Measuring productivity

Management inevitably wants KPIs



Management inevitably wants KPIs

Lines of Code

Issues/backlog closed

Number of commits

Story Points

```
private bool IsEven(int number){  
    if (number == 1) return false;  
    else if (number == 2) return true;  
    else if (number == 3) return false;  
    else if (number == 4) return true;  
    else if (number == 5) return false;  
    else if (number == 6) return true;  
    else if (number == 7) return false;  
    else if (number == 8) return true;  
    else if (number == 9) return false;  
    else if (number == 10) return true;  
    else if (number == 11) return false;  
    else if (number == 12) return true;  
    else if (number == 13) return false;  
    else if (number == 14) return true;  
    else if (number == 15) return false;  
    else if (number == 16) return true;  
    else if (number == 17) return false;  
    else if (number == 18) return true;  
    else if (number == 19) return false;  
    else if (number == 20) return true;  
    else if (number == 21) return false;  
    else if (number == 22) return true;
```

SPACE framework

2021 ACM Queue, Volume 19, Issue 1

<https://queue.acm.org/detail.cfm?id=3454124k>

Satisfaction and well-being

Performance

Activity

Communication and collaboration

Efficiency and flow

FIGURE 1: EXAMPLE METRICS

LEVEL	SATISFACTION & WELL-BEING How fulfilled, happy, and healthy one is	PERFORMANCE An outcome of a process	ACTIVITY The count of actions or outputs	COMMUNICATION & COLLABORATION How people talk and work together	EFFICIENCY & FLOW Doing work with minimal delays or interruptions
INDIVIDUAL One person	<ul style="list-style-type: none"> * Developer satisfaction * Retention† * Satisfaction with code reviews assigned * Perception of code reviews 	<ul style="list-style-type: none"> * Code review velocity 	<ul style="list-style-type: none"> * Number of code reviews completed * Coding time * # Commits * Lines of code† 	<ul style="list-style-type: none"> * Code review score (quality or thoughtfulness) * PR merge times * Quality of meetings† * Knowledge sharing, discoverability (quality of documentation) 	<ul style="list-style-type: none"> * Code review timing * Productivity perception * Lack of interruptions
TEAM OR GROUP People that work together	<ul style="list-style-type: none"> * Developer satisfaction * Retention† 	<ul style="list-style-type: none"> * Code review velocity * Story points shipped† 	<ul style="list-style-type: none"> * # Story points completed† 	<ul style="list-style-type: none"> * PR merge times * Quality of meetings† * Knowledge sharing or discoverability (quality of documentation) 	<ul style="list-style-type: none"> * Code review timing * Handoffs
SYSTEM End-to-end work through a system (like a development pipeline)	<ul style="list-style-type: none"> * Satisfaction with engineering system (e.g., CI/CD pipeline) 	<ul style="list-style-type: none"> * Code review velocity * Code review (acceptance rate) * Customer satisfaction * Reliability (uptime) 	<ul style="list-style-type: none"> * Frequency of deployments 	<ul style="list-style-type: none"> * Knowledge sharing, discoverability (quality of documentation) 	<ul style="list-style-type: none"> * Code review timing * Velocity/flow through the system

† Use these metrics with (even more) caution – they can proxy more things.

DORA (DevOps Research and Assessments)

See <https://dora.dev>

"DORA is the largest and longest running research program of its kind, that seeks to understand the capabilities that drive software delivery and operations performance."



Image source: Hadian Rahmat

<https://medium.com/gits-apps-insight/dora-metrics-how-to-measure-software-delivery-performance-e890ec2011c0>

DORA pitfalls (<https://dora.dev/guides/dora-metrics-four-keys/>)

- Setting metrics as a goal
- Having one metric to rule them all.
- Using industry as a shield against improving.
- Making disparate comparisons.
- Having siloed ownership.
- Competing.
- Focusing on measurement at the expense of improvement.

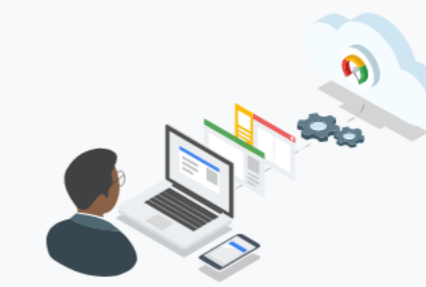
DORA quick check → <https://dora.dev/quickcheck/>



[Take the 2024 DORA Survey now!](#)

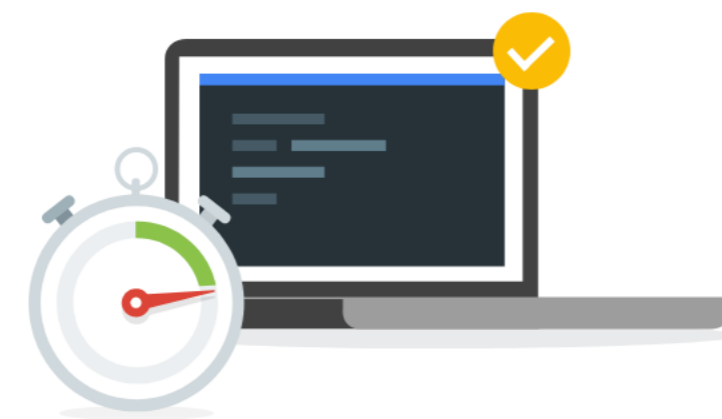
Take the DORA Quick Check

Measure your team's software delivery performance in less than a minute! Compare it to the rest of the industry by responding to four multiple-choice questions. Compare your team's performance to others, and discover which capabilities you should focus on improving. We don't store your answers or personal information.



QUESTION 1 OF 4

Lead time



For the primary application or service you work on, what is your **lead time for changes** (that is, how long does it take to go from code committed to code successfully running in production)?

- More than six months
- One to six months
- One week to one month
- One day to one week
- Less than one day
- Less than one hour

QUESTION 2 OF 4

Deploy frequency



For the primary application or service you work on, **how often does your organization deploy code** to production or release it to end users?

- Less than once per six months
- Between once per month and once every six months
- Between once per week and once per month
- Between once per day and once per week
- Between once per hour and once per day

Your software delivery performance

Compare to industry benchmark:

Your performance

5.5

Lead time

One to six months

Deploy frequency

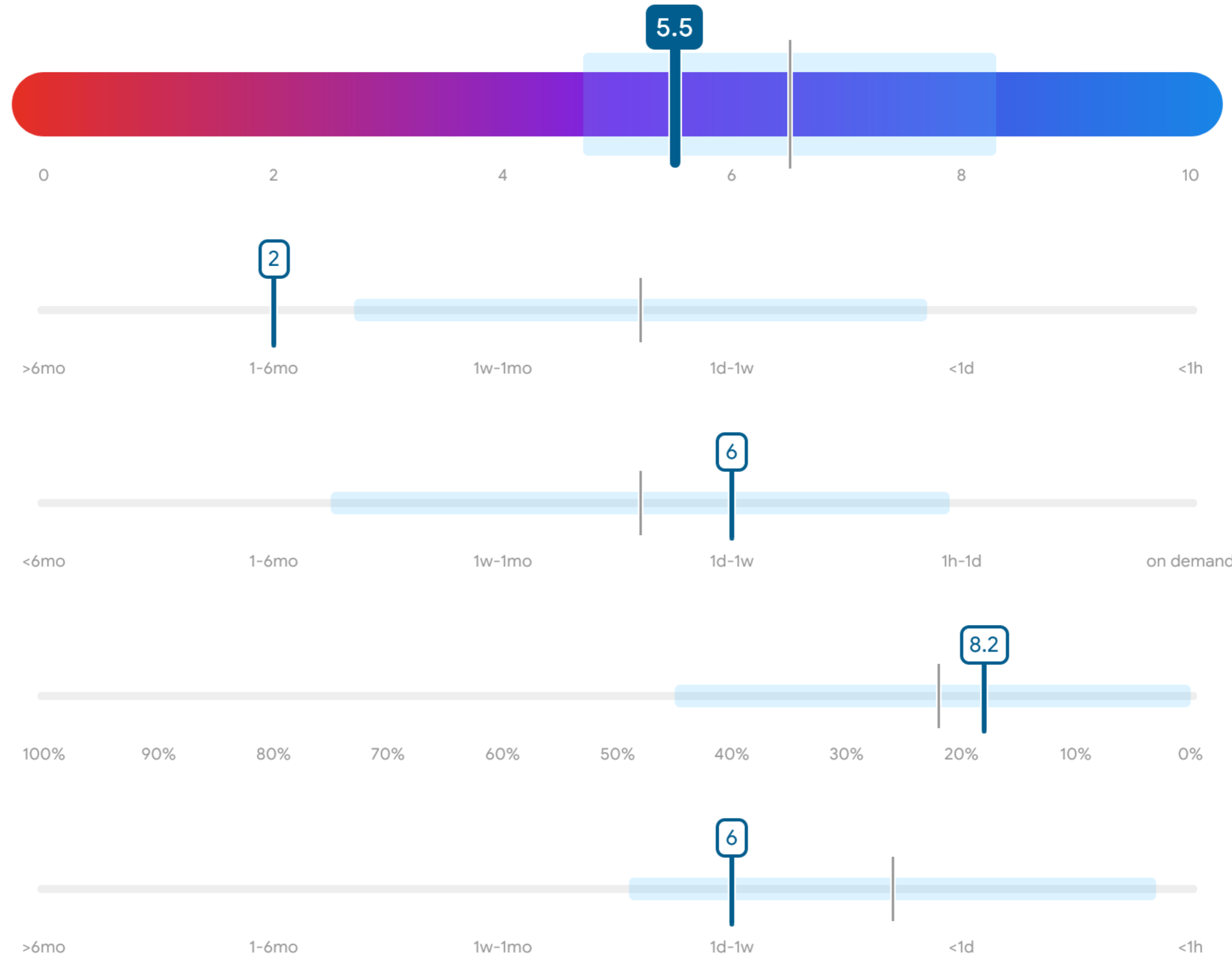
Between once per day and once per week

Change fail rate

18% of changes fail

Failed deployment recovery time

One day to one week



2023 Industry baseline (Financial Services):

| Average

Standard deviation

Your performance

Abi Noda presentation at DPE summit emphasizes higher focus on qualitative metrics

<https://dpe.org/sessions/abi-noda/were-measuring-productivity-wrong/>

Quantitative metric	Qualitative metric
PR cycle time	I work on small, iterative changes. <i>Never</i> <i>Rarely</i> <i>Sometimes</i> <i>Very often</i> <i>Always</i>
Commit frequency	I have uninterrupted time for deep work. <i>Never</i> <i>Rarely</i> <i>Sometimes</i> <i>Very often</i> <i>Always</i>
Time to first review	I receive code reviews in a timely manner. <i>Never</i> <i>Rarely</i> <i>Sometimes</i> <i>Very often</i> <i>Always</i>

Building 10x developers



10X ENGINEER



Myth Origin (probably) The Coding War Games





Origin (probably) - The Coding War Games

- In 1977, authors Tom DeMarcos and Timothy Lister devised a study called the **Coding War Games**
- From 1984 – 1986, more than **600 developers from 92 companies** participated
- The purpose was to **discover the “best” and “worst”** traits of developers
- The competition unit was a **group of competing programmers from the same organization**
- A program specification was fixed, and **participants logged their time** in completing it
- Participants **used their own workspace** using familiar tooling and languages

<https://www.gwern.net/docs/cs/algorithm/2001-demarco-peopleware-whymeasureperformance.pdf>

The “best” programmers outperformed the worst by roughly a 10:1 ratio

There were some interesting “non-factors”:

Language

Years of Experience

Number of Defects

Salary



What Mattered?

- Paired programmers from the same organizations **performed at roughly the same level**
- The average **difference was only 21%** between paired participants
- They didn't work together on the task, but they **came from the same organization**
- **The best organizations performed 11.1x better than the worst**

“While this productivity differential among programmers is understandable, there is also a 10 to 1 difference in productivity among software organizations.”

-Harlan (HD) Mills, Software Productivity in the Enterprise

https://trace.tennessee.edu/cgi/viewcontent.cgi?article=1010&context=utk_harlan

The best performers are clustering in some organizations while the worst performers are clustering in others.

Some companies are doing a lot worse than others.

Something about their environment and corporate culture is failing to attract and keep good people or is making it impossible for even good people to work effectively.

Average performance of those in the top quarter was 2.6 times better than that of those in the bottom quarter.

Table 8.3
Environments of the Best and Worst Performers
in the Coding War Games

Environmental Factor	Those Who Performed in 1st Quartile	Those Who Performed in 4th Quartile
1. How much dedicated work space do you have?	78 sq. ft.	46 sq. ft.
2. Is it acceptably quiet?	57% yes	29% yes
3. Is it acceptably private?	62% yes	19% yes
4. Can you silence your phone?	52% yes	10% yes
5. Can you divert your calls?	76% yes	19% yes
6. Do people often interrupt you needlessly?	38% yes	76% yes

Though the phrase had not yet been coined, increased productivity came down to developer experience.

... But Most Organizations Aren't Aligned



In a study dated April 27, 2022, between Microsoft and the University of Victoria in British Columbia, Developers and Managers were surveyed on their interpretation of the SPACE framework

When surveyed with the following questions, Developers and Managers answered much differently

Developers

When thinking about your work, how do you define productivity?

Managers

When thinking about your team, how do you define productivity?

	ICs define own productivity	Managers define team's productivity	
S	8%	9%	
P	35%	67% (*)	
A	50%	21% (*)	
C	24%	33%	
E	38%	45%	

DevOps, 12-Factor, Agile, etc, have still not captured all bottlenecks, friction, and obstacles to throughput

Many are hiding in plain sight, in the developer experience itself

A 10x organization should think about reducing build and test feedback times, and improving the consistency and reliability of builds

Calendar Today < > **May 2023**

WED 25 GMT-08

8 AM

9 AM **Code**

10 AM **Wait Time for Local Build**

11 AM **Debug Build Failure**

12 PM **Lunch / Elden Ring**

1 PM **Code**

2 PM **Wait Time for Local Build**

3 PM **Sprint** **Waiting time for CI Build**

4 PM **Investigate/Fix Flaky Tests**

5 PM

6 PM

7 PM

March 2020 < >

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

Search for people

Other calendars + ^

THE #1 PROGRAMMER EXCUSE
FOR LEGITIMATELY SLACKING OFF:

"MY CODE'S COMPILING."

HEY! GET BACK
TO WORK!

COMPILING!

OH. CARRY ON.



THE #1 PROGRAMMER EXCUSE
FOR LEGITIMATELY SLACKING OFF:

"MY CODE'S ~~COMPILING.~~"

Testing

HEY! GET BACK
TO WORK!

COMPILING!

OH. CARRY ON.



THE #1 PROGRAMMER EXCUSE
FOR LEGITIMATELY SLACKING OFF:

"CHAT GPT IS DOWN"

HEY! GET BACK
TO WORK!

CHAT GPT IS STILL DOWN

OH. CARRY ON.





















NETFLIX



Netflix reduced a 62-minute test cycle time down to just under 5 minutes!

What can you do?

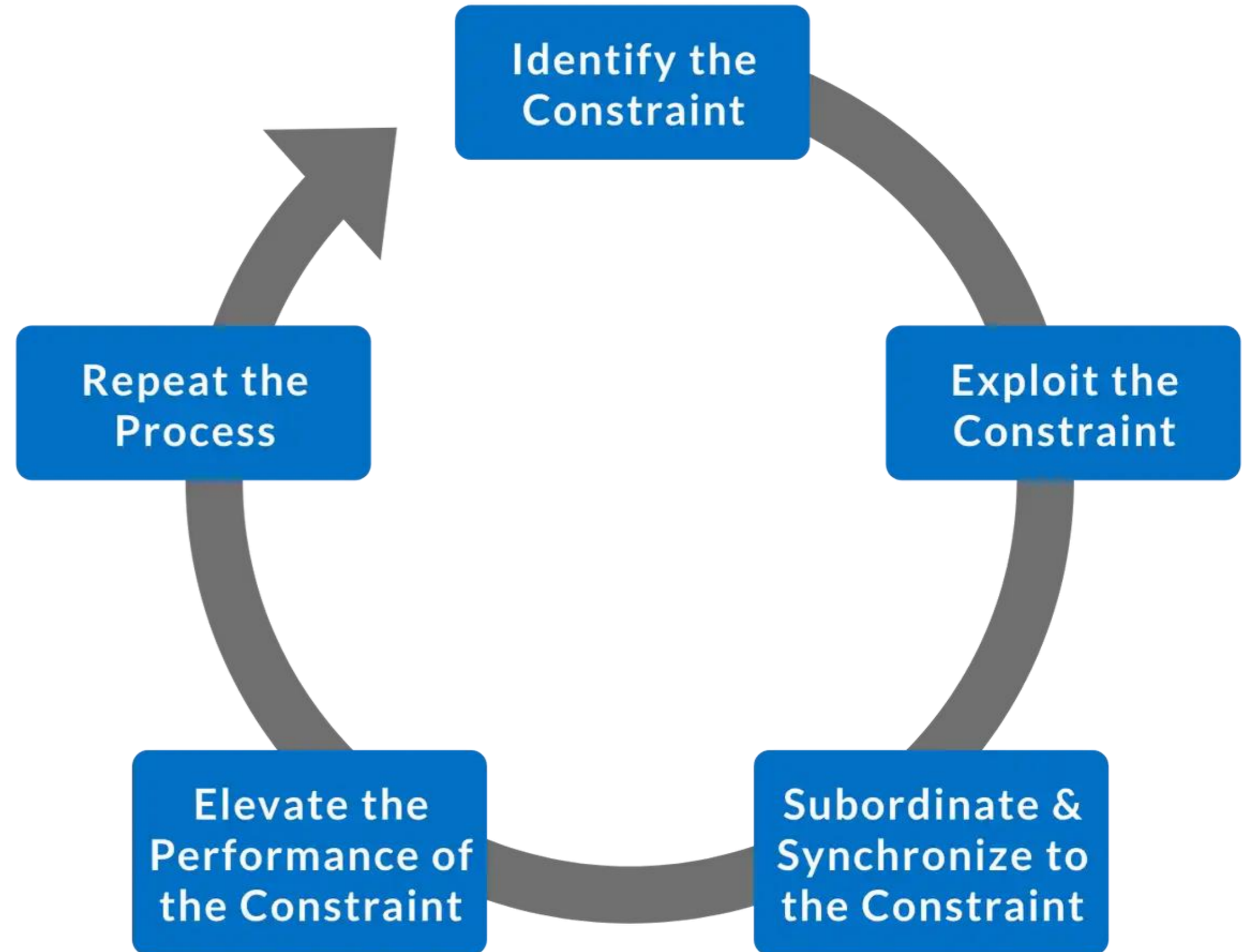
Can Therbligs apply to software development?

- | | |
|---|---|
|  Search |  Use |
|  Find |  Disassemble |
|  Select |  Inspect |
|  Grasp |  Preposition |
|  Hold |  Release Load |
|  Transport Loaded |  Unavoidable Delay |
|  Transport Empty |  Avoidable Delay |
|  Position |  Plan |
|  Assemble |  Rest |

Identify (and address!) productivity bottlenecks

Back to Theory of Constraints

<https://www.leanproduction.com/theory-of-constraints/>



Philosophical nature of constraints

Constraint	Description
Physical	Typically equipment, but can also be other tangible items, such as material shortages, lack of people, or lack of space.
Policy	Required or recommended ways of working. May be informal (e.g., described to new employees as “how things are done here”). Examples include company procedures (e.g., how lot sizes are calculated, bonus plans, overtime policy), union contracts (e.g., a contract that prohibits cross-training), or government regulations (e.g., mandated breaks).
Paradigm	Deeply engrained beliefs or habits. For example, the belief that “we must always keep our equipment running to lower the manufacturing cost per piece”. A close relative of the policy constraint.
Market	Occurs when production capacity exceeds sales (the external marketplace is constraining throughput). If there is an effective ongoing application of the Theory of Constraints, eventually the constraint is likely to move to the marketplace.

Nature of constraints applied to software engineering

Constraint	Description
Physical	Inadequate equipment Lack of space Noisy work area Lack of DevOps
Policy	Excessive meetings Company won't allow adoption of new tools
Paradigm	SEU "It's always been done this way" Siloed development teams
Market	Writing code that doesn't matter (or, at least, the developer doesn't know why it matters)

Are your problems technology problems or HR problems?

Technology
Problems

Connectivity issues

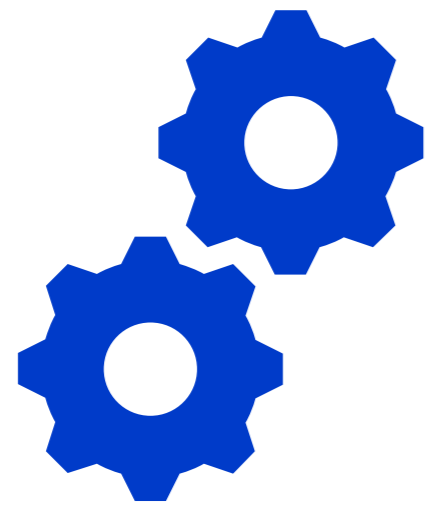
Archaic codebase

HR
problems

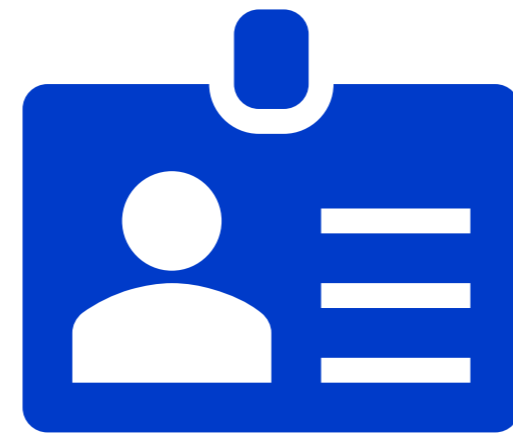
Resistance to change

Hardened processes

Modern toolchains



DevOps and CI/CD



IDE



AI

LIAMA > QRPGLSRC > EMPLOYEES.SQLRPGLE > LoadSubfile

```

106 Dcl-Proc LoadSubfile;
107   Dcl-S iCount  int(5);
108   Dcl-S Action  Char(1);
109   Dcl-S LongAct Char(3);
110
111   ClearSubfile();
112
113   EXEC SQL DECLARE empCur CURSOR FOR
114   ... SELECT EMPNO, FIRSTNME, LASTNAME, JOB
115   ... FROM SAMPLE.EMPLOYEE
116   ... WHERE WORKDEPT = :DEPTNO;
117
118   EXEC SQL OPEN empCur;
119
120   if (sqlstate = '00000');
121
122   dou (sqlstate <> '00000');
123     EXEC SQL
124     ... FETCH NEXT FROM empCur
125     ... INTO :Employee.EMPNO,

```

More Actions...

- 💡 Run statement in Db2 for i

```

1 SELECT EMPNO, FIRSTNME, LASTNAME, JOB
2 FROM EMPLOYEE
3 WHERE WORKDEPT = :DEPTNO

```

RESULTS

View will be active when a statement is executed.



workload

runner.pgm.rpgle db.sqlrpgle

```
rpgle > db.sqlrpgle > ...
462 dcl-proc updateCacheBalance;
500 | | | | | INSERT (BALCUSNO, CUSBALANCE)
501 | | | | | VALUES (SOURCE.BALCUSNO, SOURCE.CUSBALANCE);
502 | | | | | endsl;
503 | | | | | end-proc;
504
505 // procedure to get the latest balance for a customer
506 dcl-proc getCachedBalance export;
507 | | | | | dcl-pi *n zoned(10:2); You, 2 days ago • Uncommitted changes
| | | | | type char(1) const;
| | | | | cusno int(10) const;
| | | | | end-pi;
|
| | | | | dcl-f balances qualified usropn usage(*input);
| | | | | dcl-ds balance likerec(balances.BALFMT);
|
| | | | | select;
| | | | | when (type = RLA);
| | | | | OPEN balances;
| | | | | chain (cusno) balances.BALFMT balance;
| | | | | CLOSE balances;
| | | | | return balance.CUSBALANCE;
|
| | | | | when (type = SQL);
| | | | | EXEC SQL
| | | | | SELECT CUSBALANCE
| | | | | INTO :balance.CUSBALANCE
| | | | | FROM BALANCES
| | | | | WHERE BALCUSNO = :cusno
| | | | | ORDER BY BALASOF DESC
| | | | | FETCH FIRST ROW ONLY;
|
| | | | | if (sqlstate = '00000');
| | | | | return balance.CUSBALANCE;
| | | | | endif;
| | | | | endsl;
|
| | | | | return 0;
| | | | | end-proc;
508 |
509 // TODO: batchUpdateBalances()
```

WORKLOAD

- .evfevent
- .vscode
 - actions.json
- cmd
 - runner.cmd
- rpgle
 - data.rpgle
 - db.sqlrpgle
 - runner.pgm.rpgle
- sql
 - balances.table
 - cards.table
 - customer.table
 - means.table
 - trans.table
- .gitignore
- makefile

PROJECT EXPLORER

main 0 0 0 Live Share Ln 507, Col 1 Spaces: 2 LF RPGLE

PIECES COPILOT

Understanding dept.pgm.sqlrpgle

GPT3.5 16k

Generate a responsive web page based on emps.dspf and the corresponding python backend

To generate a responsive web page based on the emps.dspf file and the corresponding Python backend, you can follow these steps:

- Set up a Python web framework: Choose a Python web framework like Flask or Django to build the backend of your web application. Install the necessary dependencies and set up a basic project structure.
- Create a route for the web page: In your Python backend, define a route that will handle requests for the web page. For example, using Flask, you can define a route like this:

```
python
1 from flask import Flask, render_template
2
3 app = Flask(__name__)
4
5 @app.route('/emps')
6 def emps_page():
7     # Logic to fetch data from the database and pass it to the template
8     # ...
9     return render_template('emps.html', employees=employees)
```

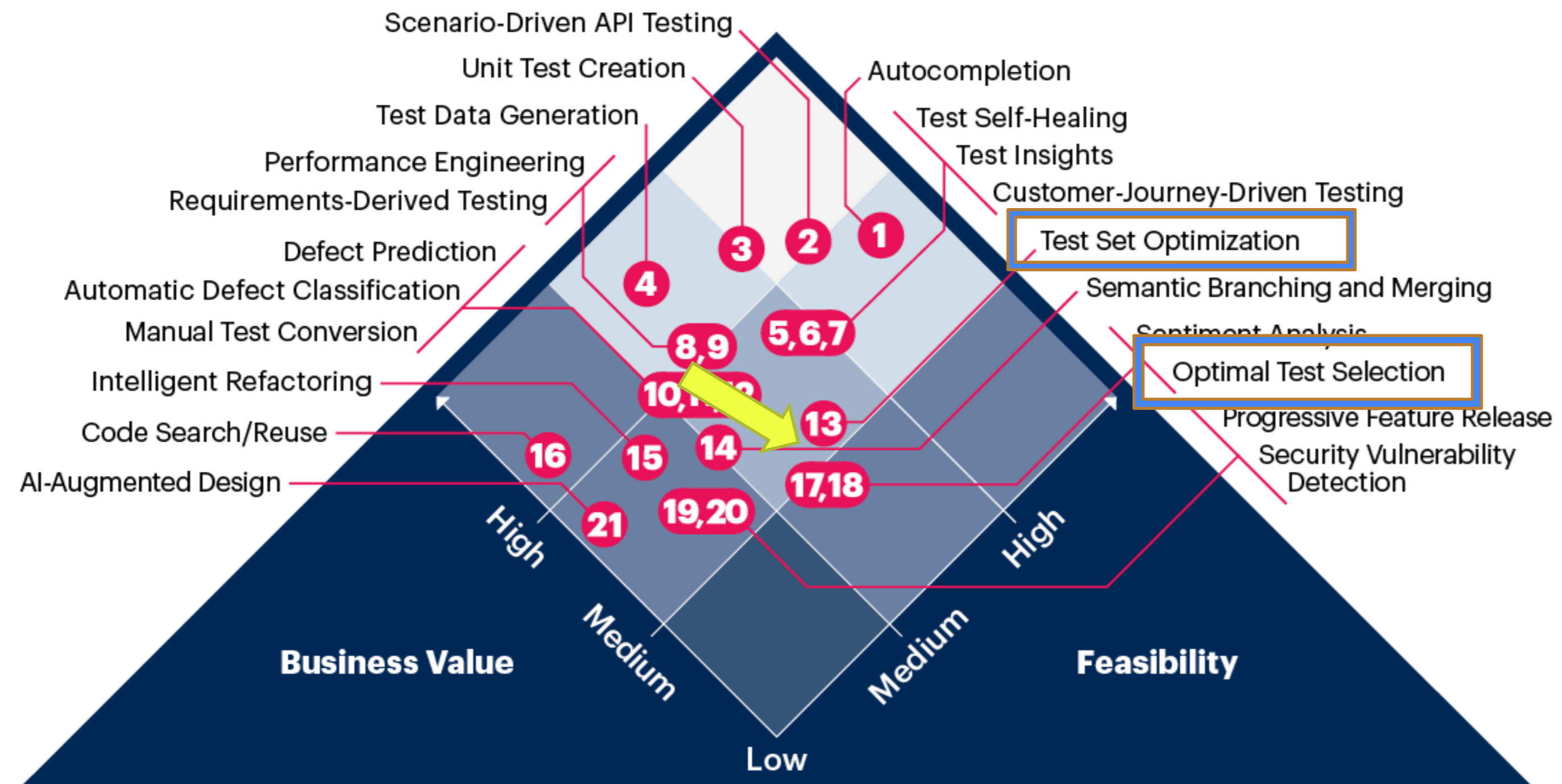
- Fetch data from the database: Write the necessary code in your Python backend to fetch data from the database. You can use a database library like SQLAlchemy or psycopg2 to interact with the database. Retrieve

Suggested Queries

What database system is being used to store the employee data? Are there any specific security measures implemented when fetching data from the database? How are potential errors or exceptions handled in the Python backend code? Are there any...

```
makefile
1
2 BIN_LIB=CMPSYS
3 LIBLIST=$(BIN_LIB)
4 SHELL=/QOpenSys/usr/bin/qsh
5
6 all: depts.pgm.sqlrpgle employees.pgm.sqlrpgle mypgm.pgm.rpgle
7
8 ## Targets
9
10 depts.pgm.sqlrpgle: depts.dspf department.table
11 employees.pgm.sqlrpgle: emps.dspf employee.table
12 mypgm.pgm.rpgle: constants.rpgleinc
13
14 ## Rules
15
16 %.pgm.sqlrpgle: qrpglesrc/%.pgm.sqlrpgle
17     system -s "CHGATR OBJ('$<' ) ATR(*CCSID) VALUE(1252)"
18     liblist -a $(LIBLIST);\
19     system "CRTSQLRPGI OBJ($(BIN_LIB)/$*) SRCSTMF('$<' ) COMMIT(*NONE) DBGVIEW(*SOURCE)
20     @touch $@
21
22 %.pgm.rpgle: qrpglesrc/%.pgm.rpgle
23     liblist -a $(LIBLIST);\
24     system "CRTBNDRPG PGM($(BIN_LIB)/$*) SRCSTMF('$<' ) OPTION(*EVENTF) DBGVIEW(*SOURCE)
25     @touch $@
26
27 %.dspf: qddssrc/%.dspf
28     -system -qi "CRTSRCPF FILE($(BIN_LIB)/QDDSSRC) RCDLEN(112)"
29     system "CPYFRMSTMF FROMSTMF('./qddssrc/$*.dspf') TOMBR('/QSYS.lib/$(BIN_LIB).lib/Q
30     system -s "CRTDSPF FILE($(BIN_LIB)/$*) SRCFILE($(BIN_LIB)/QDDSSRC) SRCMBR($*)"
31     @touch $@
32
33 %.table: qddssrc/%.table
34     liblist -c $(BIN_LIB);\
35     system "RUNSQLSTM SRCSTMF('$<' ) COMMIT(*NONE)"
```

AI Use Case Prism for Software Development and Testing



Source: Gartner: Artificial Intelligence Use Case Prism for Software Development and Testing
<https://www.gartner.com/document/3994888>

Industry-standard technologies in your solution



Organizational factors

Change agents

Remote work?

Excessive meetings?

Performance reviews

Management focus on enablement

Community involvement

Your homework

Resources in this handout

Do a DPE-centric analysis when:

- Managing a team
- Collaborating with peers
- Performing

Promote developer satisfaction by:

- Identifying pain points
- Addressing directly

BE A CHANGE AGENT!!!

**DPE Will Become Standard Practice
Because the World Should Foster Developer Joy**

Thank you!!