

The Critical Path

How I Find the Limit
Before the User Reaches It

Ioan Marian

QUOTES AND ESSENTIAL LESSONS



ONE METHOD IN THREE DIRECTIONS

From incident to continuity

I understand what is happening, I reduce unnecessary work, and I build so that the system can continue.

The 42 ideas follow this path.

01

I Understand What Is Happening

I start with the effect, I look for the cause, and I build the milestones that I can verify.

01 In the Middle of the Storm · 02 Before I Build, I Look for the Limit · 03 The Log Is the First Line of the Application · 04 Benchmarks and the Breaking Point

02

I Remove Unnecessary Work

I keep the same result, change the path up to it and measure again.

05 The Critical Path · 06 One Million Rows and the Choice of Another Path · 07 The Concrete Truck Does Not Wait · 08 There Is No 30-Minute Miracle · APPENDIX Where Cost Is Born Inside a Loop

03

I Build Continuity

I look ahead, accept what I do not know, and turn experience into a method that can be passed on.

09 I Do Not Evaluate What Is Promised. I Evaluate What Endures · 10 The Patience Not to Know Yet · 11 It Worked Well. Until It Did Not · 12 When the System Can Continue Without Me

In the Middle of the Storm

01

01

One line of code blocked an entire platform

The platform was already blocked. I no longer needed to measure how slow it was. I needed to find what had stopped it.

02

By tomorrow, make it four times faster

Pressure does not force me to change a lot. It forces me to choose precisely.

03

We no longer have the documents

A backup that has never been restored is still only a promise.

Before I Build, I Look for the Limit

02

The limit

04

A test tells me that the application works. A volume test tells me how far it can go.

The purpose

05

I do not optimize for an impressive number. I remove unnecessary work so the application remains fast, easy to understand, and easy to change as data and requirements grow.

Optimization buys me time for the next increase

06

The speed gained today becomes time to react tomorrow.

The Log Is the First Line of the Application

03

The instruments

07

Without a log, it is like flying an aircraft without instruments. It works well until you reach the first storm.

The file name says what started and when

08

A good log does not merely say that the operation ended. It tells the full story of the execution.

I observe several days or read the history that already exists

09

I read the figure in context, confirm the trend, and act before it becomes an incident.

Benchmarks and the Breaking Point

04

The algorithm, input data, sorting, and result

10 My first benchmark was not a millisecond. It was a result I could verify.

One field, ten, fifty; one row, one thousand, one million

11 Good measurement starts with a case where I know exactly what I asked for.

Excel, SharePoint, identity services, PDF, and external services

12 Technology changes. The questions about volume, width, wait time, and repetition remain.

A longer time does not always mean a less efficient operation

13 Today's measured time tells me where I am. The curve tells me where I am heading.

I repeat until the trend becomes clear

14 I measure until the limit is no longer a surprise; the number is only the trace left by the measurement.

The Critical Path

05

15

The same page, but 7.65 GB, 4.38 MB, or 0.38 MB

I am not interested only in how many times faster one approach is. I want to know whether the final time remains low as the volume grows.

16

The same 101,000 rows: one second or one third of a second

Modulo 4 gives me four clear groups. Measurement tells me how much I gained. In this test, the same read finished in about one third of the time.

17

The same 300,000 results: 16.035200 seconds or 0.015229 seconds

The search can cost more than the operation performed after the result has been found.

One Million Rows and the Choice of Another Path

06

18

Ten records said yes. One million said no

I did not build the engine in five minutes. I chose the direction in which I would build it over the next nine months.

19

First the IDs. Then only the rows of the current page

I did not make the million rows easier to process. I stopped carrying them to a page that did not need them.

20

A change had to be made once, not on every page

The problem was the number of places where the same decision had to be repeated, not the number of pages.

21

The approach expanded; it was not confined to a single product

A good model does not survive being frozen. It survives being clear enough to be adaptable.

The Concrete Truck Does Not Wait

07

22

Thirty minutes for a few people. Much more for the entire construction site

The thirty minutes saved moved the work into a much harder and more expensive form.

23

A delay creates a new list of work

When I miss the moment, I do not merely add time. I add stages.

24

I do not prepare a client's application before I know the client

I prepare how I will respond well when the question becomes concrete.

There Is No 30-Minute Miracle

08

From 0.25 seconds to almost 40 seconds

25

Five percent does not look alarming on a Friday. Repeated every week for two years, it changes the nature of the problem.

From 40 seconds to approximately 0.19 seconds

26

A trend sustained over time matters more than one spectacular day.

The margin must be monitored after optimization

27

Optimization creates margin. Periodic measurement shows me when it starts to disappear.

I REMOVE UNNECESSARY WORK

APPENDIX

Where Cost Is Born Inside a Loop

A

28

The same useful work, from one instance to a billion new instances

The number of loop iterations does not tell me how many instances are created. The position of the new instruction decides that.

29

A million values can occupy 32 MB or 4 MB

The same values can have very different costs depending on how they are laid out in memory.

30

I increase the volume tenfold until the difference becomes clear

The breaking point is the limit I encounter in a test before the application encounters it.

I Do Not Evaluate What Is Promised. I Evaluate What Endures

09

I measure the small operation and follow where its effect goes

31

I look at the small operation first. Then I track how often it repeats and how far its effect reaches.

I build scenarios, not prophecies

32

I can recognize a structure that leaves no room for growth even when the future is not known exactly.

I ask not only whether it will endure, but also what will break first

33

I look for the signal that appears before the incident.

The Patience Not to Know Yet

10

34

It is a starting point, not an excuse

“I do not know yet” is professional only if I can tell what I am going to measure next.

35

The page is slower, but I still do not know why

Patience does not delay the decision. It prevents a decision based on the wrong cause.

36

My model is not reality

When reality contradicts my model, I improve the model.

It Worked Well. Until It Did Not

11

The Protection Could Be Forgotten

- 37** If safety depends on my remembering every time, I have no protection. I have only a hope.

The fact that it worked ten times does not make it safe

- 38** A risky deviation repeated without incident does not become correct. It becomes familiar.

I would no longer leave the protection to my own attention

- 39** Experience helps me recognize the risk. Automatic protection helps me when attention lapses.

When the System Can Continue Without Me 12

40

What is shared must be nameable

A good model allows someone else to guess less and verify faster.

41

AI can build and multiply a good model very quickly

AI multiplies the model. I still decide what is worth multiplying.

42

A system that someone else can continue keeps the method, not the person

Another person can operate, investigate, and modify the system without making the author an indispensable dependency.

PUBLIC CODE AND RESULTS

<https://github.com/loanMarian2001/DemoCriticalPath>

Exact version used by the book: book-v1.0.0