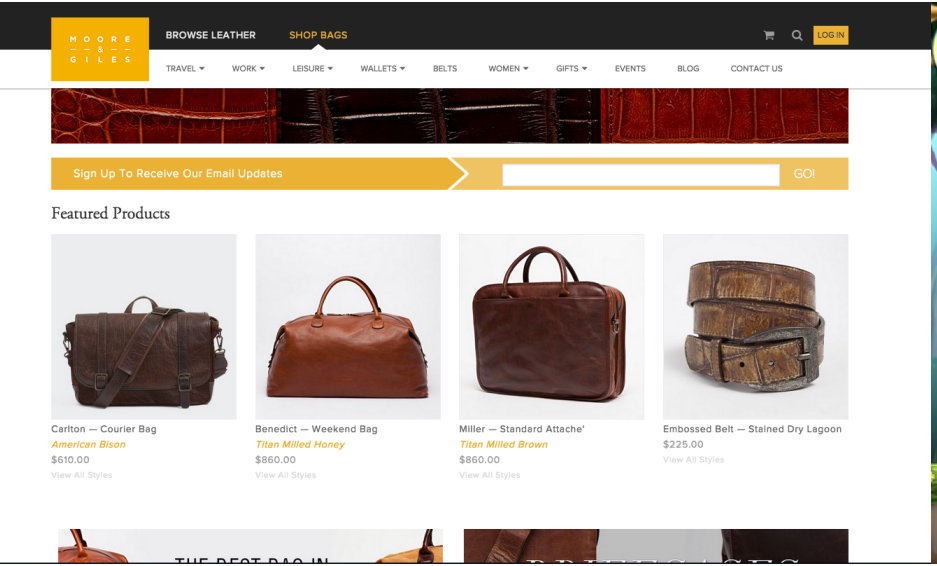
How Users Perceive the Speed of The Web

Paul Irish, Google Chrome
FluentConf 2015

there’s this word…

Slow

What is slow?

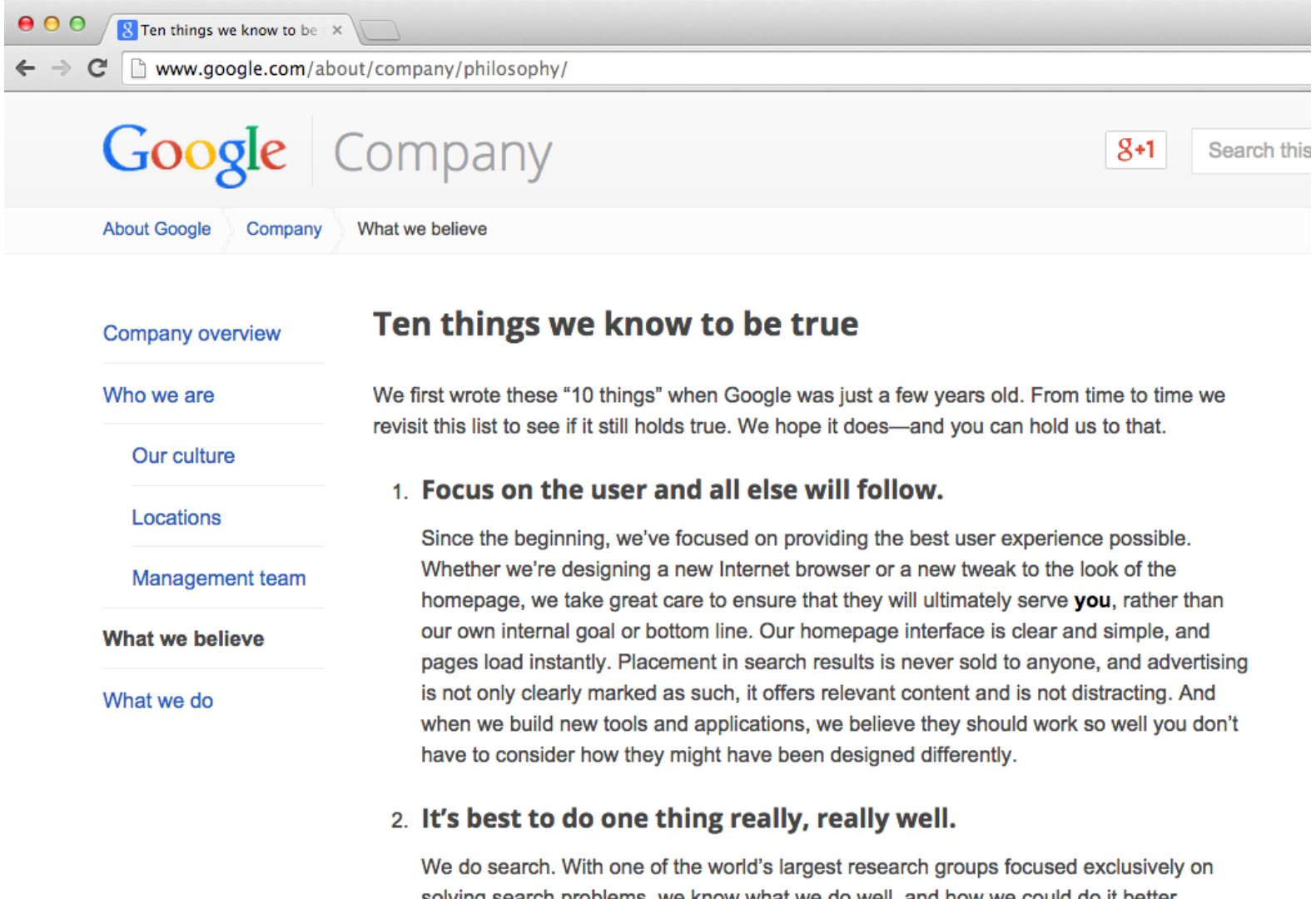

MOORE & GILES
BROWSE LEATHER
SHOP BAGS
LOG IN
TRAVEL
WORK
LEISURE
WALLETS
BELTS
WOMEN
GIFTS
EVENTS
BLOG
CONTACT US
Sign Up To Receive Our Email Updates
GO!
Featured Products
Carlton — Courier Bag
American Bison
$610.00
View All Styles
Benedict — Weekend Bag
Titan Milled Honey
$860.00
View All Styles
Miller — Standard Attache'
Titan Milled Brown
$860.00
View All Styles
Embossed Belt — Stained Dry Lagoon
$225.00
View All Styles


120
PIXIE POWER!

We need metrics!
We need a budget!

What metric matters?

- DOMContentLoaded?
- FPS?
- Jank?
- First paint?
- Sunspider?


Ten things we know to be
www.google.com/about/company/philosophy/
Google
Company
g+1
Search this
About Google
Company
What we believe
Company overview
Who we are
Our culture
Locations
Management team
What we believe
What we do
Ten things we know to be true
We first wrote these “10 things” when Google was just a few years old. From time to time we revisit this list to see if it still holds true. We hope it does—and you can hold us to that.
1. Focus on the user and all else will follow.
Since the beginning, we’ve focused on providing the best user experience possible. Whether we’re designing a new Internet browser or a new tweak to the look of the homepage, we take great care to ensure that they will ultimately serve you, rather than our own internal goal or bottom line. Our homepage interface is clear and simple, and pages load instantly. Placement in search results is never sold to anyone, and advertising is not only clearly marked as such, it offers relevant content and is not distracting. And when we build new tools and applications, we believe they should work so well you don’t have to consider how they might have been designed differently.
2. It’s best to do one thing really, really well.
We do search. With one of the world’s largest research groups focused exclusively on

FOCUS ON THE USER AND ALL ELSE WILL FOLLOW

What is slow?

~~What is slow?~~

What does the user feel?

In 1993,

based on research from 1968 & 1991

100ms

1000ms

10 seconds

Response Time Limits: Jakob Nielsen (1993)

100ms, 1000ms, 10s

100ms gives the feeling of **instantaneous** response

Results feel immediate.

Any longer and the connection between action and reaction is broken.

1000ms keeps the user's flow of thought **seamless**.

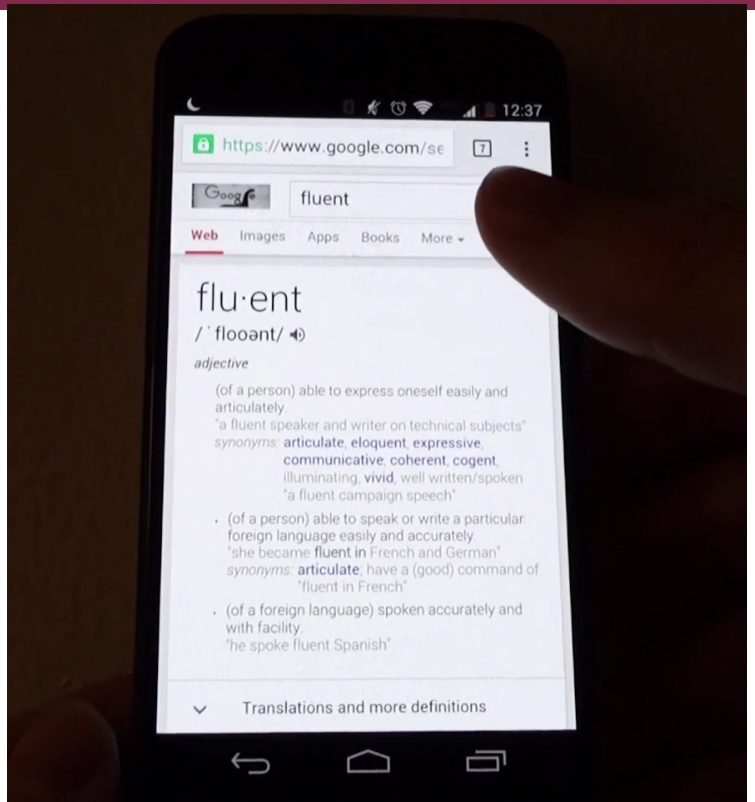
Things feel part of a natural and continuous progression of tasks.

Beyond it, the user loses focus and attention.

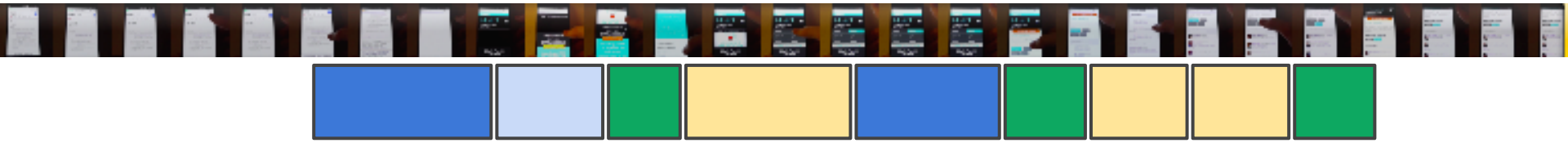
Beyond **10 seconds** you've lost the user's **attention**.

Let’s look at the user’s journey

View video

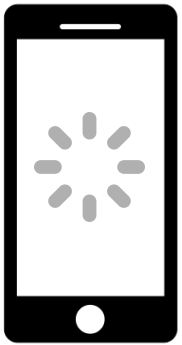


What does the user feel?

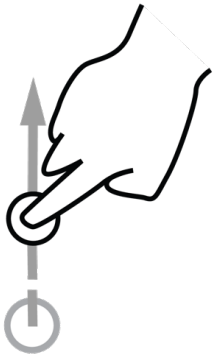


Four phases of interaction

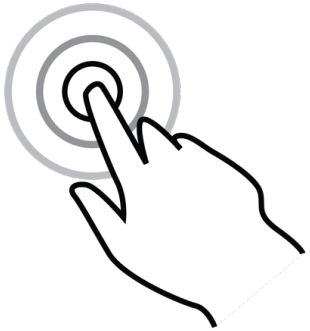
Page load



Idle



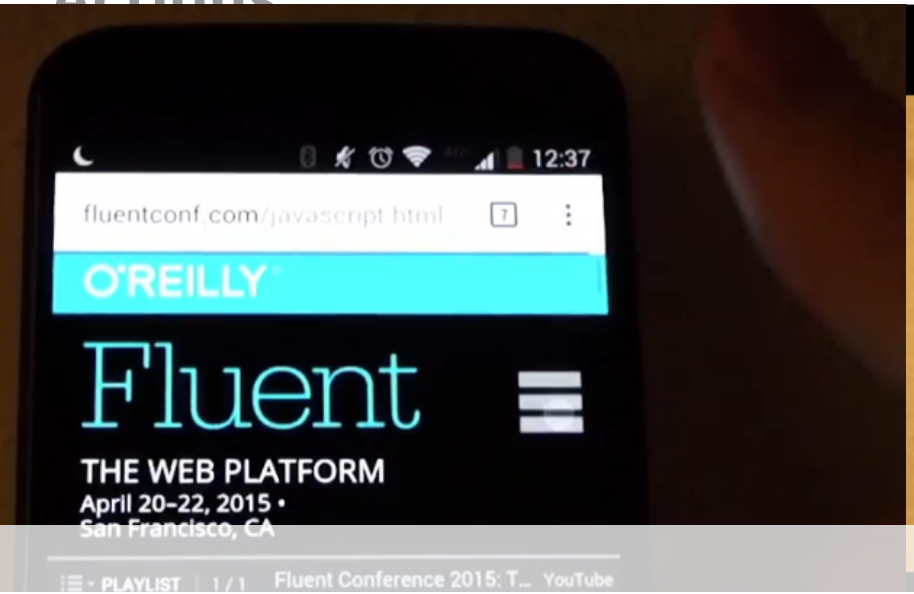
Response to Input

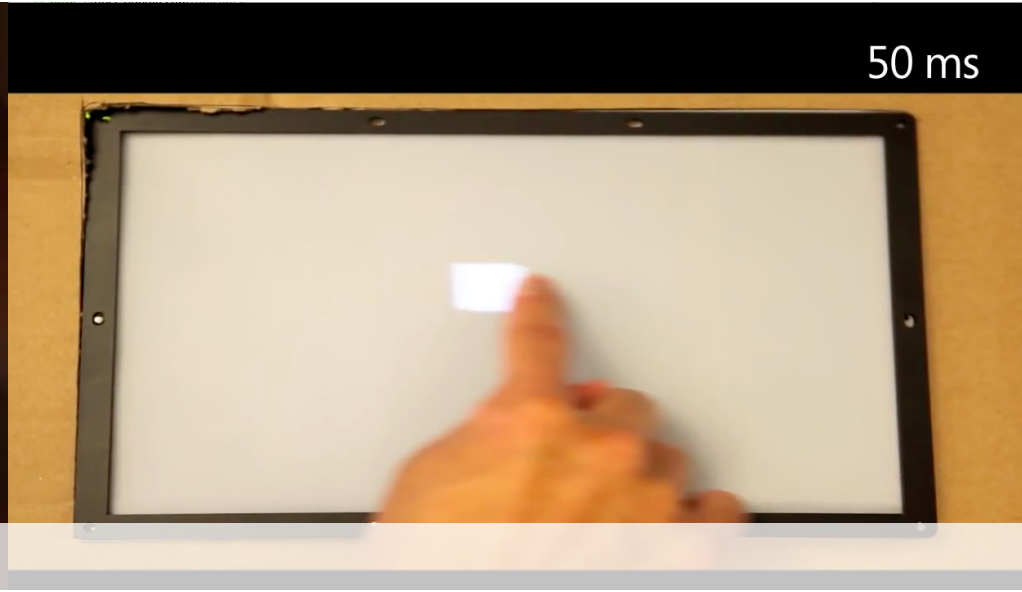


Scrolling & Animation

Response

Actions:





The **input latency** from **tap to paint** is short.

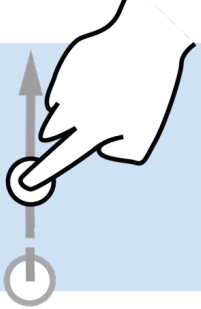
Animation & Scrolling

Actions:

User sees an animation

User scrolls the page

Full-fidelity **60 FPS** smoothness

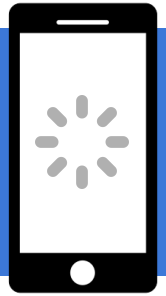


Idle

Action:

User isn’t active right now… but could be.

Page is **ready for interaction** when needed



Load

Actions:

User loads the page and sees the critical path content.

User loads the page and starts interacting.

🐶 The **page is considered ready to interact** with quickly.

Response to Input

Animation & Scrolling

Idle

Page Load

Response

Animation

Idle

Load

RAIL

Performance Model

What’s the target goal?

RAIL performance goals

Response

Tap to paint is < 100ms

Touchmove to paint is < 16ms

Animation

Each frame completes in < 16 ms

Idle

Use idle time to proactively schedule work

Complete that work in 50ms chunks

Load

Ready to use in 1000ms

Satisfy the response goals during full load

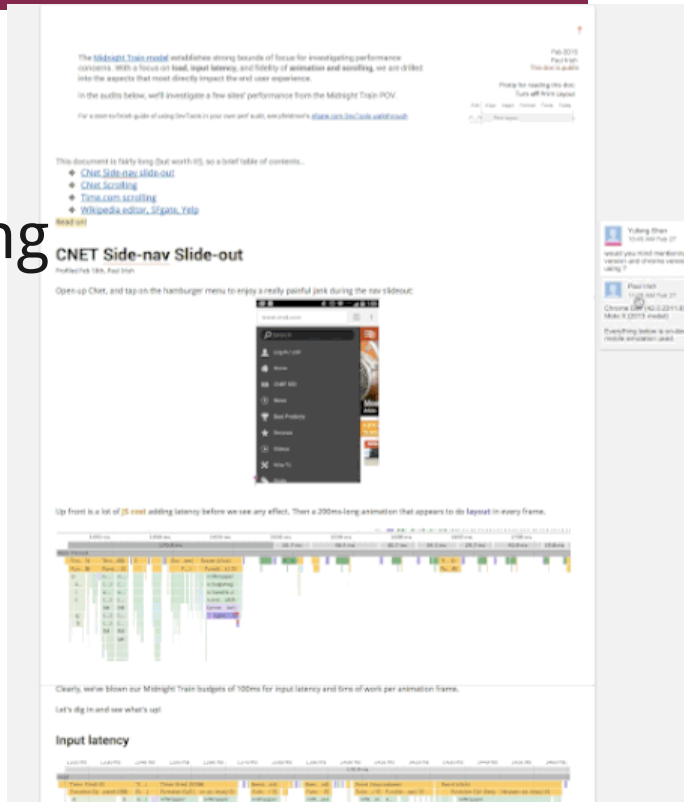
Have a question on performance?

RAIL is a place to start the conversation

RAIL Performance Audits

on paulirish.com

- **CNet**: input latency & scrolling
- **Time.com**: scrolling
- **Google Play:** infinite scroll
- **Wikipedia**: webapp startup
- **ESPN**: pageload, input
- *more coming...*



RAIL is a work in progress…

We want your feedback & ideas

Thanks!

@paul_irish
Google Chrome

Recommended performance budgets

Milestone timings

e.g.

- Time to top headline
- Time to inspected element
- Time to first tweet

Speed Index

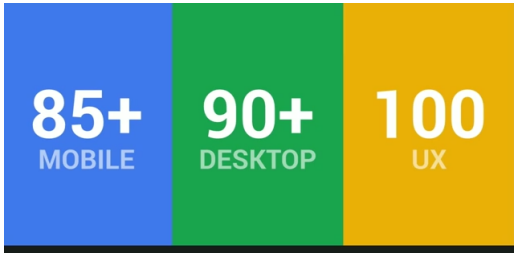
2500

Pro-level:

1000

Rule-based metrics

PageSpeed Insights:



It started as "Midnight Train"

Midnight Train: Developer Edition

Midnight Train: Developer Edition

File Edit View Insert Format Tools Table Add-ons Help Last edit was made yesterday at 6:18 PM by Paul Lewis

Authors: Paul Irish, Paul Lewis
Jan 15, 2015

Let's try to take go/midnight-train & bit.ly/midnight-train and translate them to the web developer audience. There is a gap between this performance model and how web developers have traditionally thought. But developers will absolutely jump on this bandwagon if we can successfully connect the historical UX research → their perspective → the timing guidelines. It's gonna be awesome.

First up a big matrix of characteristics and brainstorming, then timing goals, then next steps…

	Initial load	Quiescence	User-initiated UI state change	Input-bound effect	Transition from one UI state to another
	transition from start state → the state in which the app appears to be available to the user. the full browser-side path	the app UI is in a stable state, awaiting user's command	the user interacted with the app, causing it to start changing the UI state	the user interacted with the app, and its UI feedback was synchronized to the input	whether user-initiated or autonomous, the app changes the state of the UI.
Threshold / Guideline	1000ms	unknown ! be conscious of its impact on device battery, heat, and memory consumption.	100ms max	6ms chunks == 60fps	6ms chunks == 60fps (needs to accept that 16ms includes browser overhead)
current developer vocabulary	pageload	idle	click handler, event handler	touchmove	transition / animation (css & JS) / scrolling
relevant vocabulary	loading	non-critical, cancellable / pausable work	input latency, responsivity, feedback,	input latency, {touch input scroll} -linked, user-driven,	Visual fidelity, fire and forget, jank free

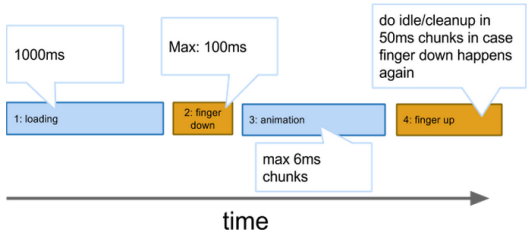
https://docs.google.com/a/google.com/document/?authuser=0&usp=docs_web

This is Success: Why 1000/100/6/50ms?

Platform Success Model Explainer

nduca@, dglazkov@
bit.ly/blink-midnight-train

We have previously presented that the following application lifecycle is what we want to enable:



But, why?

The guidelines

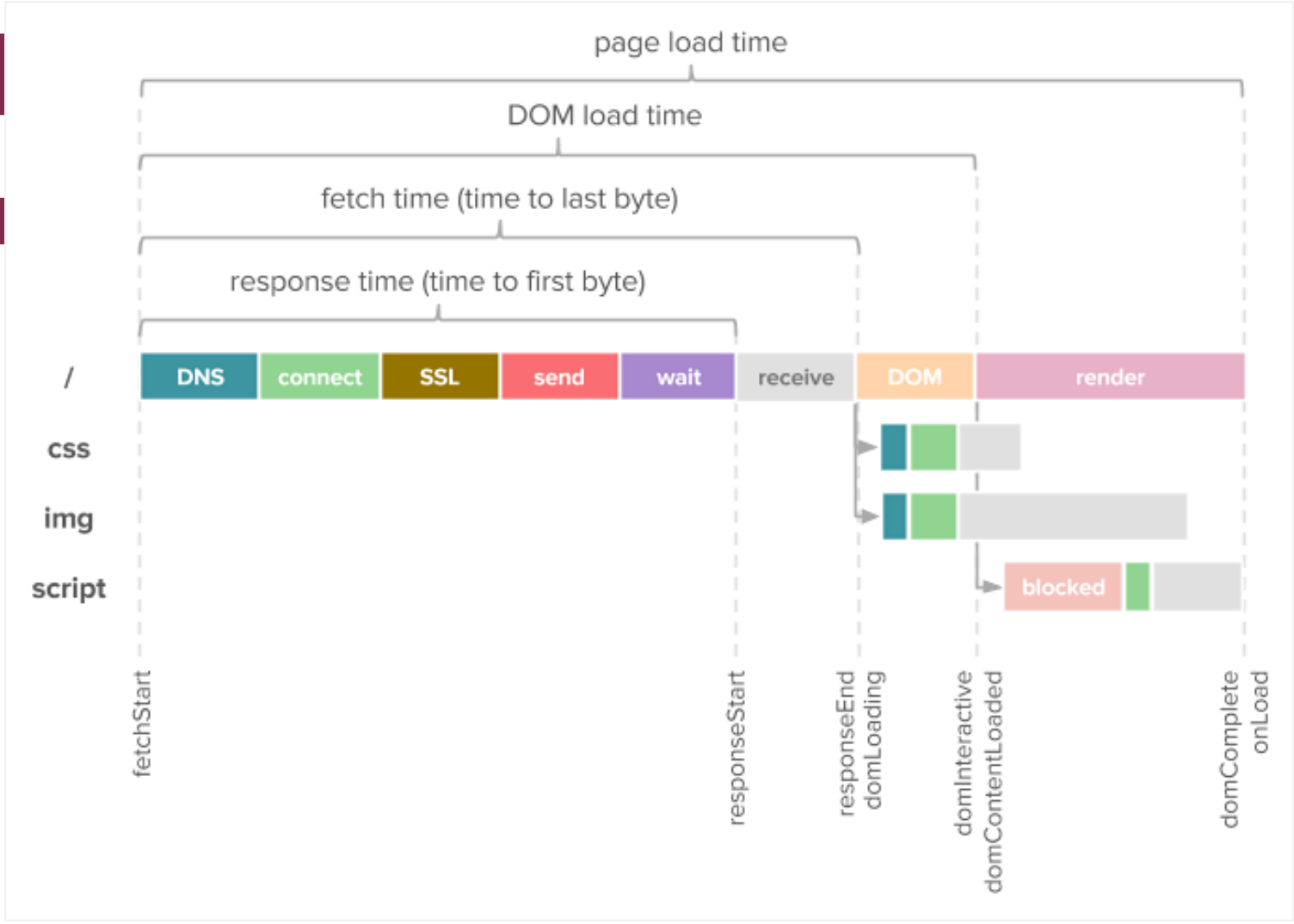
We can view developer's ability to make a great app if our system obeys the diagram above.

- **User productivity is not impacted by response times below 150 milliseconds.** This is therefore a good quantitative definition of crisp response.

- In the range from **150 ms to one second, users become increasingly aware of response time.** They strongly prefer response times well below one second.

- **Above one second, users become unhappy.** When forced, users can adapt to response times over one second, but this is accompanied by frustration with the system and a drop in productivity

W


page load time
DOM load time
fetch time (time to last byte)
response time (time to first byte)
/
DNS
connect
SSL
send
wait
receive
DOM
render
css
img
script
blocked
fetchStart
responseStart
responseEnd
domLoading
domInteractive
domContentLoaded
domComplete
onLoad

— NOVEMBER 18, 2014 —

Performance Budget Metrics

1. Milestone timings ✔
2. Speed Index ✔
3. Rule based metrics ✔
4. Quantity based metrics

For the purposes of this post, I'm breaking those metrics down into four categories:

1. Milestone timings
2. SpeedIndex

Performance advice abounds :/

Don’t use jQuery :hidden

Don’t use * selectors

Try/catch forces deopts

JS animation slower than CSS

translate3d() for the zoom!

Event delegation is good

Event delegation is terrible

Tell me more, JSPerf!

Don’t use with() statement

Avoid for..in loops