Introduction
Good afternoon everyone! Thank you for having me, my name is Garret Rempel. I am a Senior Technology Consultant with MNP from Canada. I have been with my current employer for almost 10 years, and I have been working in the technology industry for almost 20. I have also been involved with performance analysis, investigation, and resolution for a large part of the last 8 years.

The paper I wrote is on Defining Standards for Web Page Performance in Business Applications. And really, what it is about is the poor quality of industry standards that exist today for web page performance, and a way by which we can go about defining performance requirements that are reasonable, useful, and are done in collaboration with the people who will be responsible for achieving them.
My Job
On a new project, my job often begins with educating my clients. I work with their business and technology experts to aid them in building their strategy for setting performance targets, testing their applications, measuring and monitoring performance, and resolving any issues discovered.
Problem
One of the first problems I typically encounter on any new project though, is that, **people**, are *terrible* at setting performance targets.
And developers are not really any better. People are *awful* at assessing their own patience level and just as bad at translating that into actual requirements. Can you consider, the one person, that I have been given to talk to about setting performance requirements, feels that waiting 30 seconds for a web page to load is just fine?
Challenge
So the challenge is, how do we set accurate, and precise performance requirements in a manner that involves input from both the business and the developers in order to ensure they are invested in those requirements? In a way that we have buy-in from all parties involved?

If the business doesn’t believe in the target, then they may undermine us with the end user. If the developers don’t believe in the target, then they won’t put a priority on fixing things when we fall short.
Industry Standards

So! If we look towards industry literature for a gold standard to follow, what we find are results that are varied, inconsistent, based on questionable methodology. Frankly there is limited consensus and about as many ways of studying the subject as there are studies. Many recent ones actually rely on their subject’s own self-assessment which, let’s say, is of questionable value.
In fact, some of the most useful information dates back to Miller in 1968 who was studying Human-Computer Interfaces. Today, what we are seeing is a convergence and transparency between online and offline modes. People often don’t know whether the information they are accessing is local or remote, and they are starting to expect the same level of responsiveness regardless. They expect that what they want is at their fingertips instantaneously, and they aren’t going to give you the same leeway that they used to, just because you are providing an online service.

What we do know is that response times under 0.1s a person perceives as being instantaneous. Unfortunately if we are operating in a client-server model, consistent response times under 0.1s are probably not achievable – considering it takes light 0.13s to circle the earth. We start running into limitations imposed by physics. Response times under 1s though are fast enough for a person to maintain continuity of thought. That means that they do not perceive any time spent waiting, the response arrives as fast as their continuous thought process is able to interpret it. It is only in the range of 2-4s where a person begins to perceive the fact that they are actually waiting for something to happen. And once a user perceives that they are waiting, that is when interruptions, breaks in their train of thought, and
frustration can start to take hold.
Of more recent results the most referenced studies are able to provide some additional valuable insights, such as providing a responsive interface with feedback if the user needs to wait will actually result in users patiently waiting twice as long or more before becoming frustrated. However these studies also use one of the more useless gauges from a business perspective – median abandonment.

Tell me. If I am presenting to the technology director of a Fortune 500 company using one of these studies as the basis for my recommendations – and I state that if we work to achieve this level of performance I can say with confidence that only HALF of your customers will abandon your website out of frustration with its performance – really, what good is that?

If we are to set an industry standard, it must be one that produces a high level of satisfaction among users with a minimal, or better yet, zero rate of abandonment.
Case Study

In order to better understand this abandonment threshold I had the good fortune to be able to work with a client to study the performance of a system I had previously helped them with. A system that had been in production mode for some time, and then take this knowledge and apply it to a new project. The system I studied has the following profile:

- Industry-leading client information tracking and incident reporting system supplied by a reputable international vendor
- 1200 users across 5 time zones in Canada and the United States who use the system as part of their primary duties
- Peak usage is 800 simultaneous login, 50,000 page requests per hour over a 4 hour window.
- Average weekday receives 440,000 page requests with peak of 510,000 on the busiest day of the week, and 10,000,000 per month.

The observation period of the study is a span of almost 2 years starting from initial go-live, a period of decaying performance, optimization, and then stable operation.
The advantage of studying this particular system is that this is a closed ecosystem. The users have no recourse to abandon the application, and they have no alternatives to work around the problems they encounter. They are motivated and encouraged to report problems and incidents of poor performance through a centralized service desk.
Case Study Results
What we saw was a clear correlation between aggregate page response times, and the number of performance complaints that were issued to service desk. As you might expect, slower performance means more complaints.

What is very interesting to me though, is that the magnitude of the change in response times was really quite small, but effected a significant change on the number of complaints. And the performance threshold needed to reach a level of 0 complaints is rather tighter than many of the previously mentioned industry papers.
The most noticeable difference between the phases of the application lifecycle is at the 95\textsuperscript{th} percentile mark, which hovers around a threshold of 4-5s during the initial stages, improves to under 2.5s during its stable period.

What these numbers suggest is that for this class of application, for these users, in order to have an acceptable level of performance we need:

• 95\% of page requests on aggregate must be completed within 2.5s
Proposal
Our goal is to establish a process to define performance requirements that takes advantage of business user’s input and experience and produces a result that closely matches our case study observations.

So in order to match our case study we must have:
• 95% of all web page requests achieve end-to-end response times of 2.5s or less
• A majority of individual pages achieve a target of 2s or less
• And, we need to allow for a limited number of pages to have targets greater than 2.5s where achieving that level of performance would be infeasible. These pages will be identified as candidates to provide additional response feedback in order to reduce the impact of waiting on the user’s perception of system performance.
Process
To accomplish this, we designed a set of page performance categories with pre-set performance targets and in collaboration with system developers, we set specific examples of pages that would fit into each category.

A page is considered to have achieved its performance target when under typical load, the percentile response time measurement for the category meets the target, and under peak load the percentile response time meets the maximum.

The business users were then brought together and asked to categorize the complete list of pages and workflow functions into one of the given categories, or to note an exception and provide justification for why the exception was necessary.
Results

Using this process against the application in the case study, we identified 259 distinct web pages or workflow functions that were categorized and evaluated.

85% of the pages were placed by the business into the Basic Operations category with a target of <2s. 11% were in the next category of Complex Operations, given a target of <5s. And only 8 pages, or 3%, were given larger targets. Every page was placed by the business into one of the predefined categories with not a single exception.

Calculating the weighted average for all the page performance requirements produced a resulting Target Response Time of 2.56s for 94% of all pages, and a Maximum Response Target of 3.14s.

This is pretty close to our actual observed level of 95% of pages at <2.5s that resulted in a 0 complaint level, not quite, but pretty close.
But it gets better.

When we adjust the categorized pages for their expected frequency of use what we see is that the Basic Operations category increases from 85% to almost 90% of all page requests.

Taking the weighted average then results in a Target Response time of 2.37s for 94.3% of all page requests, and a Maximum Response Target of 2.69s.

If our application can hit these targets, then we are right in line with what our case study has shown us, that if we can get 95% of all page requests under 2.5s, our application will perform at an acceptable level for everyone using it.
Conclusions
What this shows us is that having a general understanding of our target audience’s wait time tolerance, with some assistance from the development team, we can enable business users with no particular training or experience in the field of performance, to independently define requirements that closely align with their user’s needs.
Slides:
1. Background
2. Problem Statement
3. Convergence in Expectations
4. Types of Studies
   1. Physiological Measurements
   2. Empirical Studies
   3. Empirical Studies – Feedback vs None
   4. Surveys
   5. Findings
5. Case Study
   1. System Under Scrutiny
   2. Methodology
   3. Findings
      1. Communication Latency
      2. Render Time
      3. Response Percentiles
      4. Complaints
      5. Major Percentiles
6. Gathering Requirements
1. Categorization
2. Process
3. Page Aggregates
4. Request Aggregates

1. Conclusions