

## **Audio Transcript of Steven Cerri's Engineer-to-Leader Podcast Episode 8: " Micromanagement Is Easy To Avoid... Really!" (Part 1 or 2)**

Hello, you're listening to The Engineer To Leader Podcast, Episode Number Eight. Today we're talking all about micromanagement, what it is, and how to avoid it. This is part one of a two part series, so stay tuned.

Hi, I'm Steven Cerri and I'm here to help you advance your engineering, scientific and technical career by showing you how to avoid the mistakes and the missteps that can slow down or even derail your promising career. And I'm the first to say that mentors and coaches and the right advice can make all the difference in your career success and your career speed of advancement, as long as you couple it with the right mindset; the right way to think about your career path.

How do you find the right mentor, the right information, the right advice and the right mindset to put all of this together and build a career that you want? This is The Engineer To Leader Podcast.

Hello, this is Steven Cerri. Thank you for joining me on this Engineer To Leader Podcast. In this podcast, I'd like to talk to you about something that everyone seems to fear, and that is micromanagement. If you're an engineer, or a scientist, or technologist, you probably dread being micromanaged, and if you're a manager, you may be one of those managers who does not want to be called a micromanager even if you are. In my opinion, micromanagement is not to be feared and is actually no big deal. It's really easy to avoid.

Let me explain what I mean. Let's begin by asking this question. "Should you worry about micromanagement?" The answer is no. It's just not a big deal. Micromanagement is actually unnecessary and I'll show you why, and there is no reason why an employee needs to be micromanaged. (I will explain to you later on how to avoid it.) There's no reason that a manager needs to be labeled as a micromanager either. There are ways to get around this really easily. Whether you're an employee or a manager, micromanagement can be avoided, and I'll show you how.

First, let's talk about what most employees want, what most engineers, scientists and technologists want in their day to day task work. Most of us feel we want a sense of empowerment. We want to be able to make some decisions. We want to feel that we're

independent to some extent. We want to feel competent. Most of us want to feel trusted. That our manager trusts who we are and what we are doing, and most of us want to have some feeling of control over our work. Almost all of us, obviously, therefore, want to produce good work so we can have a positive performance review, whether that's annual, or biannual, whatever the situation might be.

Now, most managers want something as well from their daily work. Most managers want their employees to get their jobs done well, and most managers don't want to feel like they are babysitters. Generally speaking, most managers really don't want to feel like they're just watching over and babysitting their engineers, or scientists, or technologists. Regardless of what a lot of engineers and scientists and technologists believe, most managers don't want to babysit.

Most managers do want to control the situation well enough so that they can feel that there is going to be success. They want to feel that the employee is not going to fail, because all managers generally don't want surprises. Most managers want to avoid surprises, want to prepare for the uncertainties, and want to be given a heads up if an uncertainty shows up. Most actually, I believe, want to give positive performance reviews. Most want their employees to succeed. At the same time, most managers also want to get positive performance reviews from their managers. There's no benefit for them to watch you fail or to see you fail. They want you to be successful and most managers just don't want a lot of problems and headaches to deal with.

When you look at this, you come to one basic conclusion. No one really wants micromanagement; everybody really just wants to be successful.

But before we talk about avoiding micromanagement. Let's first talk about what it is. Let's define it, so we can have some sense of starting with a baseline definition for micromanagement.

Here's the first general definition for micromanagement that I found on the Internet. It's this:

*“Micromanagement is a management style whereby a manager closely observes or controls the work of employees. This generally has a negative connotation.”*

Before we move on to the next definition. Let me just say, I really don't know what “*closely observes or controls*” mean. That's not a quantifiable set of parameters. All right. We'll leave it at that for now.

The next definition is:

*“The micromanager is the manager who must personally make every decision. The micromanager must take a lead role in the performance of every significant task, and in extreme cases, the micromanager may dictate every small step workers take.”*

All right, fair enough, but let me ask you, it says that a “micromanager must personally make every decision.” But sometimes a manager has to make certain decisions and they may seem very minute. I'm not sure that I understand this definition very well either, and you'll see why as we move forward.

Then, there's a third definition that I'd like to put forth of micromanagement, and that is”

*“To many employees the micromanager is a control freak.”*

I don't know what that means, because control freak for one person may not be a control freak for another person.

*“The micromanager hovers over people who are trying to get their work done, and rarely seriously considers their ideas or opinions.”*

All right I understand that one.

*“The only original thinking the micromanager recognizes is his or hers.”*

All right, this is a slightly better definition. Still, I'm not really satisfied with it, but it's better than the previous definitions.

Let's look at another set of definitions for micromanagement.

*“The micromanager takes a central management practices to extremes and interferes with the employees' ability to do their jobs properly.”*

Now, this to me is a very useless definition. I don't know what “extremes” means and I don't know what “properly” means.

*“The micromanager takes management practices to extremes.”*

I don't know what that means. One person's extremes may not be somebody else's extremes and *“interferes with the ability of the employee to do their job properly”*. What does properly mean?

Here is another ambiguous definition of micromanagement that I don't find very helpful. Let's look at it.

*“Micromanagement is readily recognized by employees but most micromanagers don't think of themselves as micromanagers.”*

This is completely bogus. This is an absolutely useless definition. Most employees recognize micromanagement. That's not true. There are plenty of people who under one set of managers are absolutely fine. Other people under the same set of managers would consider themselves to be micromanagement micromanaged. This doesn't help.

Let's look at another one:

*“A boss or manager who gives excessive supervision to employees.”*

I have no idea what “excessive” means.

One final definition:

*“A micromanager, rather than telling an employee what tasks need to be accomplished and by when, will watch the employees actions closely and provide rapid criticism if the manager thinks it's necessary.”*

Well, I have to tell you, if my employee is failing or having a difficult time, I want to be able to identify it quickly, and I want to help. This definition doesn't really define micromanagement very well, either.

Now, you may be asking, Steven, why did you put all of these definitions in this podcast? Because I want you to understand that there are many different ways that people talk about micromanagement. All of them, in my opinion, aren't very useful because they're not definitive. The bottom line so far is that there's no good definition of micromanagement because they don't allow us to clearly know when micromanagement is taking place in an unambiguous fashion.

It seems to be left up to each participant, whether it's the manager or the employee to decide whether or not micromanagement is taking place and that's just not good. That doesn't help us to figure out how to deal with something if we have a thousand different people, and each one has a different definition of what micromanagement is. So this is not good.

Before we go further into micromanagement, I need to give you a few behavioral traits of what micromanagers do. Once again, I'm going to warn you that these aren't very useful definitions or useful behavioral traits.

Here's what I mean. One way of thinking about micromanagement, and one way that people think about micromanagement is the following:

*“Micromanagers resist delegating.”*

This may be true or maybe it's not. If I haven't worked with an employee I may not know how capable the employees is. I may indeed, resist delegating for a while. So I don't know what that definition really means either. That doesn't really tell me whether or not some micromanager is really a micromanager or it is just behavior that could reasonably show up because of the circumstances.

Another definition is that:

*“Micromanagers immerse themselves in overseeing the projects of others.”*

But sometimes that's absolutely necessary, so does that mean the person, the manager, is a micromanager?

Another definition, or another trait, another behavioral trait is that:

*“Micromanagers start by correcting any details instead of looking at the big picture.”*

Well, I have to tell you, most failures in projects and tasks are not because people are looking at the big picture. It's because they're not looking at the details. The idea that a manager is supposed to sit back and look at the big picture and not focus on the details, that's not a good formula for a manager.

Then finally, another behavior is that:

*“Micromanagers discourage others from making decisions without consulting them.”*

All right now this is a little better, but there's a balance here. There are going to be times when I want my employee, my engineer, scientist or technologist, if I'm a manager, to make decisions. That's cool. But there are other times when I want to make sure I'm involved in every decision because it may be very important and I don't want this person to go down a path that will ensure failure. That's also not a good sign of a micromanager.

I'll give you a couple of more behavioral traits. These in my estimation are pretty ridiculous statements, but they're out there and people use them all the time.

*“When it comes to delegation a micromanager cannot delegate effectively or delegate at all.”*

Okay, if somebody can't delegate at all, then that may be a real problem, but it may not be micromanagement.

*“A micromanager often hands out only the easy, boring or dirty tasks while delegating nothing of interest or importance.”*

Okay, maybe, maybe not.

*“A micromanager may handout work and hover providing detailed direction, dictating methods rather than providing proper preparation making the employee responsible for the results and not allowing him or her to figure anything out and learn by doing.”*

Now, this is a really interesting definition of the behavior of a micromanager and I want to unpack this a little bit because I'm going to give you some insight in both how the employee can respond and how the manager can respond, and what it looks like from a manager's perspective and from an engineer's, scientist's, or technologist's perspective.

Let me read that sentence again.

*“A micromanager may handout work and hover providing detailed direction dictating methods rather than providing proper preparation.”*

Fair enough. When a manager delegates he or she should make sure that the employee is prepared and the task that has been delegated is delegated in some complete form. Proper preparation is a good term, but then it goes on to say:

*“making the employee responsible for results and not allowing him or her to figure anything out and learn by doing.”*

Let's be really clear. If I have an important engineering, scientific, or technological task that needs to be done, I don't want my employee to be learning by doing unless there is time to allow for learning by doing and if they fail, we can pick up the pieces. The only time I'm going to let my employees learn by doing and not help them is if they have plenty of time and failure is okay. These definitions I think are really tricky and you might just want to be careful that you're not thinking in this way when you think about micromanagement.

Then finally:

*“Micromanagers may hand out a task, but pull it back at the first sign of trouble failing to provide the employee with the condition essential for growth and development.”*

That's fair enough. A manager shouldn't be pulling a task back just because things get a little sketchy. They should be helping the employee. Then it goes on to say:

*“The micromanager or manager should allow the reasonable freedom to fail.”*

Reasonable freedom to fail is an interesting phrase. I can't imagine that there are many tasks that you, as an engineer, or scientist, or technologist, are going to be working on a task where your manager is willing to tolerate your failure. That's useless. It's a ridiculous definition.

What I'm trying to do is give you an idea that across the board, there are various definitions, various philosophies of micromanagement, which make it essentially useless to define and talk about micromanagement in any concrete, reasonable fashion.

I am going to give you a definition that we can use to go forward that will allow us to talk intelligently about micromanagement and how to avoid it. We're going to get away from these ambiguous terms like: *“too much”*, and *“extreme”*, and *“learn by doing”*, and *“hover around”*, and *“reasonable freedom to fail”*, and *“resist delegating”* and *“correcting tiny details”*.

Those are just all useless terms, and we're going to go into some real concrete concepts. But I need to give you one more definition before we go into my definition of micromanagement.

A word that I want to define is “oversight”. Oversight is the amount of attention and/or information a manager wants to give and receive from the employee or the employee wants to give and receive from the manager.

We all have a sense of what oversight is, but oversight is how much information, how much contact does the manager want from the employee and how much contact does the employee want to give the manager.

An employee who wants to be left alone to do their job with little communication with the manager wants very little oversight.

An employee who wants to communicate frequently with their manager is an employee who wants a lot of oversight. This is pretty straightforward.

A manager who wants to communicate with their employees very little and wants to just give the employee a task and walk away and have employee do the work on their own, wants to provide very little oversight.

A manager who wants to know what the employee is doing all the time, or how the employee is getting along in their task and wants information frequently; that's a manager who wants a lot of oversight.

There's no right or wrong in any of this, as you're going to see in a little bit. Any one of these four situations is no better than another, and you'll see why as we move along.

Let me give you some examples of tasks and you think to yourself what level of oversight you would want. This is a mental exercise for you, who are listening to this podcast, and I'll define a situation and I'd like you to just think about how much oversight you'd be willing to tolerate from your manager if you are an engineer, scientists and technologists.

If you are a manager, just think about how much oversight you want to have of your employee if the tasks that I'm going to define for you here are what you are giving them.

Situation #1:           Imagine (as an engineer) you are given an easy task that you have done before successfully. How much oversight would you want? That's the situation number one.

Situation #2: Situation number two, imagine you are given a difficult task that you have done before successfully. How much oversight would you want now, or tolerate?" Now is it the same as the first one or is this different?

Situation #3: Let's go to number three. You're given a difficult and high-risk task that you have done before, successfully. It is difficult and high risk. Do you want the same level of oversight as before, or do you want something different? If you are an employee or a manager, how does it change or does it change at all from the previous situations?

Situation #4: Let's look at situation number four. You are given a difficult task with a short duration that you have done before successfully. It's a difficult task with a short duration. Once again, is the same level of oversight or a different level of oversight? You get to decide.

Situation #5: Let's look at number five. You're given a difficult task that you have never done before. Now, do you want the same oversight as before as in the other cases, or do you want a different level of oversight?

Situation #6: Let's look at the sixth situation. You are given a difficult and high-risk task that you've never done before. Now, how much oversight do you want? How much communication do you want with your manager if you are an employee or if you are a manager? How much communication do you want with your employee if they're doing a difficult and high-risk task that they've never done before?

Situation #7: Now let's look at the last one. You're given a difficult task with a short duration that you've never done before. Now, if you're an employee, what's the level of oversight that you'd be willing to tolerate? Or, as a manager, what level of oversight would you require in order to help an employee who's given a difficult task with a short duration that they've never done before?

Now, there's no right or wrong answer to any of these situations regarding how much oversight you would want, or as a manager, how much oversight you would require. I hope, however, that if you're an employee, if you're an engineer, or scientist, or technologist, you were willing to have different levels of oversight depending upon which of those tasks we were talking about. If you're a manager, I hope you are willing to require different levels of oversight depending upon which situation we were talking about.

The takeaway from this piece of this podcast is that oversight changes, or ought to change, depending upon the context. In other words, the task that's been given is only one piece of this

puzzle. The situation around which that task needs to be completed is the context and is an important consideration.

### **My Definition of Micromanagement**

Now, let me give you my definition of micromanagement. It's pretty straightforward, but it's very different.

Here's what I'd like to have you think about first. Imagine there are two people, two engineers, scientists or technologists, whichever you prefer, and two managers, two technical managers. We'll call the two engineers Engineer A and Engineer B. The two managers are Manager A and Manager B.

Now, it turns out that each of these managers and engineers have slightly different personalities, as you can imagine. Let me explain what I mean by that.

Imagine that **Engineer A** is a person who likes to communicate with their manager on a regular basis. They like to talk and update their manager on what they're doing and how things are going, so for them meeting with their manager regularly is what they like. They want a lot of communication and interaction with their manager.

**Engineer B**, on the other hand, is a person who likes to be left alone. Engineer B wants to do their work, be left alone, and it's fine if they never see anybody or talk to their manager during the duration of their task and they're just doing fine until the time comes that they need to deliver the task. Then when they've completed the task and they're ready to deliver it, they say, "Okay, here; here it is, my work is done." This is the way they like to communicate with their manager. "Leave me alone, let me do my task, when I'm done I'll bring it back, you'll be happy, we'll all be happy." I don't need to talk to you until my task is done. That's engineer B.

Now then we go to the managers.

**Manager A** is the kind of manager who always wants to understand how their employees are doing. Manager A is the kind of person who, while it's nothing personal, they just need to understand how you're doing. Do you need any help? Just for their peace of mind or maybe they have to update other departments or their manager, but they just want to communicate regularly and frequently with the engineers to find out how they're doing and if they need any help. This is what they like.

Then we go to **Manager B**, and Manager B is the opposite of Manager A. Manager B is the kind of manager who's either very busy or is more interested in testing an employees' abilities. **Manager B** just does not have the time to constantly have meetings or worry about how their employees are doing. They figure that if the employee needs help, they'll come to Manager B. Otherwise, go off and do your thing. I'm going to go off and do my management thing, and I'm going to disappear. When you complete your task, when you get done, come and see me. That's how Manager B likes to move to the world.

So, we have two engineers and two managers. In one of those categories, we have an engineer and a manager both of whom like a lot of communication and in the other category we have an engineer and a manager who don't want to be bothered with a lot of communication. Now, let's see what happens when you mix and match these managers and engineers.

### **Engineer-Manager Scenario Number One**

Let's look at Engineer-Manager scenario number one. Scenario number one is with Engineer A and Manager A. Engineer A wants and needs a lot of communication. Manager A wants and needs a lot of communication as well. The result is the manager wants a lot of updates and meetings to know how the employee is doing and engineer A is happy to oblige. Engineer A will consider Manager A to be an excellent manager because the manager cares about engineering A's success.

This is a perfect employee manager match, and it produces no feeling of being micromanaged, even though the oversight is high. Do you notice this? Both the manager and the engineer want a lot of oversight. They want to see each other, meet each other, and communicate frequently, so the oversight is high. There's a lot of hovering around here that some people might see as micromanagement, but this hovering around is fine with Engineer A and it's fine with Manager A. You can have a lot of hovering around with two people who are okay with that and micromanagement does not exist. So hovering is not the definition of micromanagement. Let that sink in a minute...

### **Engineer-Manager Scenario Number Two**

Now, let's look at Engineer-Manager scenario number two. Scenario number two is made up of Engineer B and Manager B. Engineer B doesn't want or need to communicate much at all. Engineer B is the engineer that says, "Leave me alone. I'll do my work and deliver it when it's done." Manager B doesn't want or need a lot of communication either. Manager B says, "Go do your thing, come back when you're done." Now the question is, what's the result of this match?

Engineer B wants to be left alone. Manager B has plenty to do and wants to test engineer B to determine how well he or she can handle the task on their own. In this case, Manager B is not going to ask for frequent updates and or status meetings and Engineer B is perfectly happy with that. This is a perfect employee manager match, and it produces no feeling of micromanagement.

So now, we have the opposite extreme. In this case, both the manager and the engineer don't want to be bothered with a lot of updates and a lot of meetings. In this case, it's a perfect match as well. The manager is doing their thing, the employee thinks, "Oh, this manager is great because the manager doesn't bother me," and so it's a perfect match and there is no micromanagement.

### **Engineer-Manager Scenario Number 3**

Now, let's look at scenario number three. Scenario number three has Engineer A who wants a lot of communication, a lot of oversight, coupled with Manager B who doesn't want a lot of communication. Can't be bothered. He is too busy doing other things and wants the employee to just go do their thing. What's the result of this match?

Engineer A wants to communicate a lot. Manager B doesn't need or want a lot of communication or updates. Engineer A will definitely feel that manager B does not care about engineer A's success or failure. You get that? Because manager B doesn't really want a lot of updates but Engineer A does, Engineer A is going to come away thinking that Manager B doesn't care. Engineer A will think manager B is an uncaring, arrogant, uninterested manager and doesn't really care about helping his or her employees be successful.

In this case, there's no feeling of micromanagement, but there is a feeling of being abandoned. In this case Engineer A doesn't feel micromanaged, he or she feels abandoned by Manager B. Engineer A thinks Manager B just doesn't care, and doesn't show any interest in helping his or her employees be successful. There's no micromanagement but there is abandonment.

### **Engineer-Manager Scenario Number 4**

Now, let's look at the last matchup. In this scenario we have Engineer B who doesn't want to communicate much at all. Remember Engineer B is the one who just wants to go do their job, leave me alone, I'll bring it back when it's done, and they're being managed by Manager A who wants to communicate a lot.

Manager A wants to communicate a lot and wants to know what's going on. Let's look at what the result is of this matchup. Engineer B wants to be left alone. Manager A wants to know how they're doing and wants to have a lot of updates, frequent updates. Any meeting or project task updates requested by Manager A will be seen to be a personal affront and a sign that Manager A does not trust Engineer B.

Because Manager A wants a lot of input and communication exchange, Engineer B is going to believe that Manager A is just terrible. Manager A is taking too much of Engineer B's time in meetings, and updates, and briefings. And all these communications are just useless. This is the employee manager matchup that produces the perception of micromanagement.

Only here does micromanagement show up; when the employee doesn't want a lot of communication and updates and the manager does.

Let's summarize then my definition of micromanagement, which you're going to see in a moment, is very different than the others that I read at the early part of this podcast.

Here's my definition of micromanagement.

***Micromanagement is nothing more than a mismatch between the level of management oversight desired by the employee, compared to the level of management oversight desired by the manager.***

Let me repeat that.

***Micromanagement is nothing more than a mismatch between the level of management oversight desired by the employee compared to the level of management oversight desired by the manager.***

Here are three scenarios.

1. If the employee wants the same desired oversight level as the manager, there's no micromanagement. It doesn't matter if it's high or low or in between, as long as both the employee and the manager want the same level of oversight, perfect, everything's fine.
2. If the employee wants much more oversight than the manager, however, the employee is going to feel abandoned. If the manager doesn't want a lot of oversight, and they want

to empower the employee, then the employee who wants a lot of oversight is going to feel abandoned that the manager doesn't care.

3. Finally, if the employee wants much less oversight than the manager, the employee feels micromanaged.

Micromanagement only shows up when there's a mismatch between the oversight expected of the manager and employee and it has to be a specific mismatch. When the mismatch is that the manager wants more oversight than the employee does then there is a sense of micromanagement.

### **Why is my definition so different than the other definitions?**

This may sound like there's not much difference compared to the standard definition I talked about earlier, but it is significantly different because now we have a framework from which to discuss micromanagement when it occurs. It's no longer some mystery concept that we only recognize when we are in it when we're in the middle of it.

It's now something that we can gauge and deal with from the employee's perspective and from the manager's perspective. Let me show you what I mean.

Avoiding being micromanaged or avoiding being a micromanager is actually all about having a discussion that defines and controls the expectations and processes of the employee and the manager at the beginning of the assignment regarding oversight. The way to defuse micromanagement is not to get all ticked off and in a huff and resist the manager or resist the employee or put pressure on the employee.

The way to get rid of micromanagement is to have a discussion early in the task development, at the beginning of the task that defines certain parameters, which I'm going to define for you in a minute, certain parameters which define the context, the situation, and the expectations, and that conversation can actually diffuse micromanagement. It doesn't matter whether there's a lot of oversight or a little bit of oversight. If the conversation occurs effectively, both parties will understand and buy into it.

I want to elaborate a little bit on why my definition is useful and what the conversations that I talked about look like. My definition really allows the manager and the employee or the team, actually, to avoid micromanagement by making adjustments based upon the task, the individual

or the team and the expectations of each. The key is to work to define a baseline of expectations that everybody can buy into.

That means that successful management and therefore successful avoidance of micromanagement is “context dependent”. What that means is, that depending upon the task and the structure of the task, it makes sense sometimes to have more oversight than less oversight.

Remember those six or seven situations I gave you as examples a while back in this podcast, and I asked you to think about the level of oversight that you would be willing to accept or as a manager you would require, those were different contexts. You've done a task before successfully. That's one context. You've never done a task before, and it's a short duration task. That's a second context. Those contexts ought to clearly show you that they may require different levels of oversight.

So we can begin to have a conversation around the task that allows us to discuss the duration, the risk, whether the employees done it before, and a number of other parameters that allow us to ultimately establish expectations of oversight that makes sense. When oversight is arbitrary, when a manager just says, "I want to see what you're doing," and doesn't give any explanation, that's not very useful. When the employee says, "I don't want to have any oversight," and doesn't explain why, "I just feel like I don't want to have anybody looking over my shoulder." That's not very useful. The only way to really get rid of micromanagement is to have a discussion about what the situation is and what makes sense for oversight. When you have that discussion, micromanagement disappears.

This brings us to the end of part one, of a two part series on micromanagement. In the next podcast, I'm going to be giving you specific examples, real world examples of how micromanagement was diffused, how it was avoided, how it was turned around once it began, and I think you're going to find these examples, these real world examples, eye opening as to how you can really deal with micromanagement in a positive way. I'll see you in the next episode.

If you would like the show notes that accompany this podcast, go to [www.engineertoleader.com](http://www.engineertoleader.com). Engineer to leader is one word and in the upper right hand corner of the homepage, you'll see a link to podcasts. Just follow the link, click on this episode and you'll find a downloadable PDF of show notes at the link.

Thank you so much for listening to this podcast. If you found this podcast useful, and I hope you have, please like, share and subscribe to the podcast and head over to iTunes and leave a rating and a review that would help so much. If you would like more important career information, just go over to my Facebook page; go to the Facebook search box and type in Steven Cerri, Engineer To Leader, all separate words. Once you're there join my Facebook group by typing in the search box, Steven Cerri, create your technical career, all separate words, and join my group because I'll be answering your specific questions right there inside my Facebook group.

Building the engineering career you want requires more than technical and operational knowledge. It takes a different mindset, a different perspective that's something that college just doesn't teach us. If you're truly interested in building your long-term career success, then go to [www.engineertoleader.com](http://www.engineertoleader.com) and engineer to leader is one word and sign up for my VIP waitlist for my upcoming course called, Building The Engineering Career You Want. I've created this program just for engineers, scientists and technologists, who want to take control of their careers so they can create the careers that bring them satisfaction and excitement.

It doesn't matter where you are in your career path, so sign up on the waitlist now to be notified the next time an opening is available. There's no commitment on your part. Just sign up and I'll notify you when there's an opening and you can decide then if you want to take the course. I'll see you in the next podcast. Be well.

*Steven Cerri*