

The role of a ServiceNow developer is extremely complicated, but mostly not just because of the platform's broad array of applications. With an easy ITSM implementation, I get a ton of questions to ask ever since I recognize my role in this process. How many more people are working on the project? Do we have a project manager, business analyst, architect, or SCRUM Master on staff? Is the client using a current system or is this a new system? The role of a ServiceNow Developer can vary greatly based on the size and context of an implementation. Indeed to experience the best as the servicenow developer; [Servicenow Training](#) will help you to put your career forward. Let's look at the more common roles that a ServiceNow Developer may experience during an ITSM implementation. Now let us discuss the servicenow developer technical roles.

Technical Roles of the Servicenow Developer

The technical roles of the servicenow developer are application implementer, developer and application architect.

Application implementer

Contrary to popular belief, the first position on the list is not that of a developer. In fact, the majority of the positions on this list have nothing to do with coding. It is not uncommon for a ServiceNow Developer to struggle to find time to write code in between all of the other non-code tasks, especially in consulting or smaller teams. An Application Implementer is one of the first and most important roles of a ServiceNow Developer. As an implementer, a developer should be familiar with the ITSM application's capabilities, configuration options, and, ideally, available plugins.

How do you turn an incident into a Knowledge Article? How do you add a Problem Workaround to an Incident? What exactly is the Standard Change Catalog, and how does it function? What if I want to have the ability to version Knowledge Articles? These are the types of questions you may be asked. A few directed reactions of "I don't know, and I will find out" probably wouldn't hurt your development firm credibility, but you should be able to answer basic questions about the ITSM applications.

Developer

[ServiceNow](#) offers an ever-expanding range of no-code and low-code options for developing processes, and their offering is truly impressive. However, when it comes down to it, much of ServiceNow still necessitates the use of code at some point. Workflows include scripted activities. Advanced scripts are used for business rules. Mail scripts are used in email notifications. You can even write custom scripts to extend ServiceNow's newest process tool, Flows.

Slinging code is just part of the platform experience. ServiceNow Developers, on the other hand, live in an odd world. Whereas the majority of JavaScript Developers use ES6, functional programming patterns, and asynchronous code, ServiceNow Developers mostly use Object Oriented patterns on ES5 with synchronous code. Because it bears so little resemblance to JS everywhere else, some people jokingly refer to the language we use as SNJS. Much of being a ServiceNow Developer entails learning to code specifically for ServiceNow. The distinction between server-side and client-side scripts takes some getting used to, and then there's determining which APIs are available in which scripts and determining when it's best to put the script down and walk away.

Application architect

When it comes to writing code, another role I frequently find myself in is that of an architect. You might think, "Oh, that's only senior developers, people who have been doing this for a while," but the truth is that I ended up in this role from the beginning. Working in-house or as a consultant, it is quite common to end up as a solo developer or on a very small team. Prepare to create your own implementation plans if this occurs.

Architects at ServiceNow spend a lot of time debating the wonderful configuration vs. customization debate. As an architect, you will be responsible for designing an implementation that mitigates upgrade risk while increasing growth velocity.

Now we will discuss the administrative roles.

Administrative roles

The administrative roles are technical writer, presenter, and troubleshooter.

Technical writer

Some projects necessitate extensive documentation, while others necessitate almost none. Documentation can take many forms, including release notes, architectural designs, functional designs, testing documentation, training documentation, and many more. I've had the pleasure of working with some people who are the technical writing equivalent of Tolstoy. Yes, the same guy who writes technical guides "for fun" despises writing documentation. But, really, that's my point: whether you like it or not, there's no substitute for strong technical writing skills. Even if you're incorporating documentation into your software comments or composing formal documentation, interacting your work in some written form or the other is an important part of being a Developer. So don't slack off on these skills - for clients who appreciate documentation, well-written documents are essentially your RSVP to the next project.

Presenter

Speaking of communication, you should also work on your presentation skills. Many consulting firms have few, if any, positions for backroom coders. Even on an in-house team, I found myself in situations where I had to present solutions to an audience. Presentation skills become much more essential to career advancement as a consultant. I've expected to spend just as much time talking about my code as I have writing it, whether delivering portions of a project kickoff, participating in sales demos, or running workshops. Good public speaking abilities are only the starting point. You'll also need to improve your ability to think quickly on your feet. I walked in with a workshop game plan on at least two occasions, only to have to completely change directions within the first few minutes. However, you do not need to start going to your local improv comedy club for practice. Simply practice being a comfortable, confident presenter, knowing your material, and genuinely caring about your customer's needs. The rest will fall into place on its own.

Troubleshooter

When it comes down to it, every developer's goal is to solve problems. Everything else we do, from code to communication to documentation, serves the ultimate goal of problem solving. Troubleshooting is at the heart of every problem we solve, whether it's a business process problem or a technical problem. This is one area where my Lean Six Sigma experience comes in handy, and I highly recommend looking into it.

Conclusion

The roles listed above are only a sample of the complexities of the [ServiceNow Developer's](#) job. One can easily argue for a variety of other roles and skills ranging from QA Tester to Project Manager. The essence of your role as a developer, as well as the expertise you undertake, is heavily influenced by the requirements of your team. The most crucial part is to be prepared to jump in and satisfy their requirements to the best of your ability.