

A game of snakes and ladders

Dr Kerry Gordon, Quintiles, Bracknell, UK

ABSTRACT

As statistical programming matures as a profession, Statistical Programmers are becoming more experienced and, as they do so, they are looking to forge careers (i.e. move up the ladder) as well as enhance their skills. The relationship between Statisticians and Statistical Programmers in an organization is likely to be strong, nevertheless this might be jeopardized by any perceived hierarchy in career status and opportunities (watch out for that snake...). In this paper, we shall explore the importance of equality in Statistician and Statistical Programmer career ladders, and discuss some potential approaches to career and skills development for Statistical Programmers.

INTRODUCTION

The development of PhUSE over the last few years is a testament to the fact that statistical programming is now an established profession in its own right. Not surprisingly, then, statistical programmers are increasingly expecting full career progression opportunities, aligned with similar opportunities available for other clinical research professionals within their organizations, especially statisticians. In section 1, we present an overview of career development paths as they are often characterized within an organization. In section 2, we explore how a strong foundation for a career in statistical programming might be established, while section 3 considers the broad range of skills which will be required for ongoing career development in statistical programming. We explore these skills further in sections 4 and 5, focusing on the role of the organization in developing its staff. Finally, in sections 6 and 7, we examine what other steps may be useful when considering career development for more senior statistical programmers.

1. CAREER LADDERS FOR STATISTICAL PROGRAMMERS

In many organizations, career development opportunities available to staff are characterized in the form of *career ladders* i.e. charts showing the employee's current position / job title and those immediately above and below in the same job function. In themselves, however, career ladders are unlikely to provide staff with enough detail to manage their career options and plan their own development, and the devil will often be in the detail. We illustrate this below by considering:

- The need to accommodate technical as well as management aspirations of staff
- The interaction between the Statistical Programming career ladder and those of other job functions
- Ingredients of a good job description.

TECHNICAL AND MANAGEMENT TRACKS

First things first – Statistical Programmers deserve a career ladder of their own. Although, at the outset, statistical programmers will be aiming simply to become proficient at their job, learning and applying the basic skills required, it may not be long (especially for high flyers) before they are seeking additional responsibilities. Different programmers will inevitably have different skills and personalities, for instance some may develop exceptional technical skills (but have moderate people skills), while others may become excellent leaders (with only moderate technical ability). It is important, therefore, to provide more than just a technical career path if you wish to cater for the strengths of all of your staff. At a minimum, you may wish to consider a technical and a management career ladder, as per the example below.

Grade	Management	Technical
.	.	.
.	.	.
33	Associate Director	Senior Statistical Programming Scientist
32	Manager	Statistical Programming Scientist
31		Senior Statistical Programmer
.		.
.		.

INTERACTION WITH OTHER JOB FUNCTIONS

Care is required when constructing such career ladders, however, as potential snakes abound. For instance, it will be important to ensure that adjacent grades overlap to some extent (e.g. if there are salary bands associated with each job grade, then the upper limit of one grade should be higher than the lower limit of the grade above). This will be needed to accommodate those who take a little longer to demonstrate they are ready for promotion. The need to ensure that the statistical programming career ladder is consistent with corresponding career ladders for related professionals is especially important. The credibility of a promotion may be readily undermined if, for instance, it coincides with the promotion of a contemporary Statistician to a higher grade. Even within statistical programming, there may be a need to ensure all sites within a global organization use the same career ladder and associated job descriptions (otherwise, for instance, a Senior Programmer in one country may not be perceived as Senior when working with another country).

JOB TITLES AND JOB DESCRIPTIONS

While career ladders indicate defined job titles, they will likely give little away regarding the required skills and expected responsibilities for each job grade and, of course, these will need to be described in individual job descriptions for each role. As for the career ladders themselves, the need for consistency in job descriptions is paramount. For instance, the job description for a particular grade might include all of the responsibilities from the previous grade as well as some additional ones (it will be the balance of responsibilities that will be expected to change).

To facilitate such consistency, it may be helpful to divide job descriptions into three sections: technical responsibilities, non-technical responsibilities and required skills/competencies. In this way, it will become clear to staff what skills they need to develop as well as which responsibilities should form the balance of their job role if they wish to be promoted. Moreover, it enables consistency between job roles across functions, for instance the non-technical responsibilities (such as team leadership) might be the same for a Senior Programmer as they are for a Senior Statistician (the job descriptions only differing in the technical aspects). This will become especially important for more senior roles, so that leadership opportunities are as available to programmers as they are to statisticians. Indeed, with largely equivalent job descriptions, this should also be a firm foundation for mutual respect between the disciplines, a respect which will be enhanced further by clearly defined project responsibilities for statisticians and programmers alike.

2. GETTING STARTED

Once parallel career ladders for programmers and statisticians are in place within an organization, new programmers joining the group will set off on their chosen career optimistically. Nevertheless, Day 1 in a new job can be daunting, especially if it's your first job following graduation (this is assumed for the remainder of this section). However, a planned and carefully constructed induction period will be the springboard for both acclimatizing the new member of staff and setting them up for a successful career. This is achieved most simply by the development of a standardized *Induction Plan*, based on standardized induction training modules. The need for standardization should be clear: to offer all new staff an equal start to their careers, to ensure consistency in understanding across the whole organization (this may be especially important for processes) and to minimize the work required in developing training materials. Such issues are particularly relevant in a global working environment.

The standard Induction Plan is likely to include several types of training, such as:

- Familiarization of job roles – gaining an understanding of own job role as well as those of other team members (e.g. Statistician, Data Manager, Medical Writer etc)
- Understanding of processes – which may include local processes (e.g. Sponsor-specific computing environment) as well as global SOPs
- Technical training – even though the new programmer may have a relevant academic degree, there will still be a need to introduce basic technical aspects of the job role, probably including extensive training in SAS®.

While recognizing that individual staff will have their own preferred learning styles (see, for example, Briggs Myers et al [1]), it is nevertheless well known that exercise-based training, where learners are offered the opportunity to test their understanding, is often more effective than lecture-style learning (see, for example, Kolb [2]). The approach to exercise-based training may be varied and innovative (see, for example, Gordon [3]) however, in this context, the use of competency-based exercises may be useful and the setting up of a *dummy project* may be especially helpful. Consider, for instance, the existence of an entire completed project (i.e. a reported clinical trial), incorporating all typical project parts and processes, alongside annotated user notes. For example, such a project might include a programming plan, all derived dataset specifications and corresponding programs/datasets, together with statistical output specifications, programs and outputs. Learners may then be guided through the project (with as little or as much help as desired) and asked to recreate some or all of the various project components. A real sense of understanding can then be assessed by comparing their documents, programs, datasets and outputs to those created originally. Additional review of SAS programming code may also prove beneficial.

PhUSE 2008

The use of a dummy project also offers the new programmer with an immediate opportunity to develop programs for a variety of clinical trial domains (e.g. disposition, demography, adverse events). Nevertheless, this exposure will need to be consolidated and expanded as the programmer develops, therefore it will be important to continue to provide a variety of project challenges e.g. different therapeutic indications to gain exposure to different types of efficacy data (such as diary card data).

3. GETTING TO THE NEXT LADDER ON THE BOARD

New programmers may be anxious to forge ahead quickly with their career however, before doing so, they will need to acquire some basic skills (and apply these effectively). And this doesn't just mean SAS, although the development of SAS skills, of course, will be of paramount importance. Other basic skills also need to be developed, many of these being in common with those needed by other team members (e.g. statisticians). The table below outlines a range of skills required by the modern statistical programming professional.

Skill type	Required skill
Technical	SAS programming
Technical	Other software
Technical	Solid understanding of statistics
Technical	Solid understanding of drug development and regulation
Technical	Clinical trials (including typical handling of various types of CRF data)
Technical	CDISC and other regulatory standards
Technical	Understanding of relevant therapeutic indication(s)
Processes	SOPs
Processes	Understanding of roles of Programmer and Statistician
Processes	Understanding of roles of other functions within project/organization
Systems	Internal organization systems (e.g. document management)
Personal	Team skills
Personal	Communication skills
Personal	Negotiation skills
Management	Basic management skills [#]

[#] Unlikely to be required for the very first step on ladder

Once these basic skills have been mastered, the now not-so-new programmer is ready to move on, but where? Whilst it may be tempting (for the programmer) to look out for the nearest step up, experience indicates that a better and more coherent approach to career development is to undertake some manager-led planning discussions. In such meetings, the programmer and their manager discuss short-term as well as long-term career options; they review job descriptions associated with roles of interest and, in particular, assess skills and experiences the programmer needs to develop in order to move in the preferred career direction. A number of management tools and techniques may be used to support this exercise, and a particularly valuable one is the so-called *Skills Gap Analysis Tool* (SGAT). This document succinctly outlines which skills (and experiences) are required to bridge the gap between successive job grades; a series of SGATs (per job level) can be extremely beneficial in determining an entire career map as well as providing direction for the first step. Of course, career progression is not a simple tick-box exercise; just because you've developed suitable programs to report a phase I interaction study doesn't mean you are competent at developing reporting programs for any type of study (e.g. a phase III, non-inferiority, oncology study). Nor is it one-sided. While programmers should expect guidance and advice from their manager, they themselves also need keep their side of the bargain, by earning the respect of their peers through effectively demonstrating their prowess at the non-technical aspects of the job role (in common with statisticians), such as team leadership. All in all, it will need to be agreed and understood at the career planning meetings that time and repeated (effective) experience will be important aspects of career progression.

In terms of aspirations and career-mindedness, just as with learning styles, everyone is different. As a manager, you may find that some staff have their whole career mapped out in your very first career planning discussion with them, while others may want to know from you where the first step should be. Some may prefer to stay ensconced in their current role (comfort zone). In all of these circumstances, it makes sense to remain flexible, and the perception of forward movement (in the approximate direction of the next natural step on the career ladder) is likely to be perceived more positively than only sideways movement, even if the final career destination is unknown.

4. SNAKES AND HOW TO AVOID THEM

An old career adage: "Never step on people on your way up the ladder; they will soon find you, when you are on your way down...". Or, to put it another way, as you forge a career, never forget the *Win/Win* principle: unless both parties

get something out of the deal (i.e. win/win), both parties will ultimately lose (lose/lose). This may be thought of as especially relevant in the pharmaceutical industry, where the community of statisticians and programmers is notoriously close.

Consider this principle in the case of a manager/employee relationship. If the employee strives for promotion “at all costs”, they will undoubtedly upset some peers along the way which, in turn, causes a problem for the manager, who will then be less inclined to promote the disruptive employee. Conversely, if the manager attempts to restrict career opportunities for the employee (compared, say, to a contemporary statistician), the employee may become demotivated and/or leave the organization (for better opportunities), either way, leaving the manager with a less effective workforce.

Indeed, when it comes to promotions, the win/win principle can be taken even further. While equality of status with relevant counterparts (e.g. statisticians) will be crucial (i.e. no perceived hierarchy between professions), fairness will also be vitally important to both the employee and the rest of the department. Consider a programmer recently promoted to “Senior”, for instance. From their point of view, their promotion needs to coincide with a clear increase in responsibility (and skill level), otherwise they may become disgruntled that their promotion was meaningless. From the perspective of the rest of the department, even if the employee’s responsibility does indeed increase, they need to be credible in their new role; this can often mean that the employee is capable of demonstrating sufficient emotional intelligence [4] as well as technical skill. Again, then, the win/win principle holds and promotions may become problems unless they are right for both the employee and the department.

MANAGEMENT VERSUS TECHNICAL TRACK

Early on in a career, as programmers pick up the basic skills, promotions are associated primarily with enhanced technical skills leading to enhanced project responsibilities. It would be easy to continue this trend for more senior promotions, however for those wishing to pursue a management path, this would be a mistake. The best technical programmers do not always make the best managers. Moreover, even those programmers with strong technical and people skills will benefit from a standardized management training program before they get too far along a management career path.

PROGRAMMER INVOLVEMENT

While statisticians have ready excuses for getting involved early in studies (e.g. to contribute to study design and protocol development), the value of the programmer’s input at an early stage is often overlooked. Programmers should take it on themselves to demonstrate their value, by making strong contributions at the outset of a study (e.g. CRF review, analysis plan/table shell design, programming strategy) and by demonstrating strong leadership (including proactive communication) throughout the study.

TRAINING, TRAINING, TRAINING

As mentioned earlier, everyone has their own learning style. Nevertheless, to develop a meaningful career, one cannot overestimate the importance of an ability and willingness to respond positively to feedback and to strive to learn and improve on a continuous basis. “You learn most by your mistakes” may well be a familiar maxim, but there’s no doubting the truth of it. In the next section, we shall explore the three most common types of learning in further detail: formal training, manager-led training and hands-on training.

5. ACQUIRING SKILLS

So now your programmer is up-and-running – they have learned the basic skills (and had a chance to practice these), considered their career options and have an idea of which direction they wish to take. After evaluation of the appropriate SGAT, it becomes clear which new skills they need to develop (and which existing skills they need to enhance). With the programmer’s preferred learning style in mind, the manager will now formulate a training and development strategy to get the programmer where they want to be.

FORMAL TRAINING

As a manager, you occasionally hear staff complain that they are unable to undertake an activity because they have not had formal training; in my view, this is a common misconception. In fact, research shows that only around 10% of meaningful development occurs as a result of formal training [5]. Nevertheless, such training does have its place, and is clearly preferred by some learners. Consideration should be given to the value of learning in group classes as contrasted with self-study (possibly e-Learning) – though the latter may be more cost-effective, and more transportable across regional divides, such training tends to be more static (i.e. less driven by practical experience of course delegates) and so may also be a little less effective. Certainly, personal and management skills training will often benefit from a range of delegate experiences, as would basic technical courses. Specialized technical courses (including SAS and/or CDISC certification), however, for which general experience is relatively limited, may well suit a self-learning approach.

MANAGER-LED TRAINING

Around 20% of development arises from manager-led training. This could simply involve managers and mentors demonstrating certain techniques to their staff, or may be more integrated into the culture of the workplace. Examples of the latter include:

PhUSE 2008

- Providing regular performance feedback to staff (not just at formal appraisal meetings)
- Holding *Lessons Learned* meetings at the end of each project, and ensuring the feedback is cascaded through to the entire organization
- Systematic programming code review, especially for less senior staff
- Sharing of best practice:
 - Within the programming team
 - Between sites and regions (when working as part of a global group)
 - Between functions within an organization
 - Between organizations (if politics allow!)

HANDS-ON TRAINING

The very large majority of meaningful development occurs as a result of hands-on training. In the statistical programming setting, the predominant approach tends to be to expose programmers to untried tasks directly, with suitable oversight. It might be considered that this is a slightly labor-intensive and potentially risky approach (depending on the importance of the assigned project), especially early on in a programmer's career. As mentioned in section 2, a dummy project can be very useful in this respect. However, experience shows that wide exposure to a variety of new real-project challenges is an effective way of developing staff skills at a fast pace. Clearly, this approach requires a great deal of mentoring (including programming code review), however the investment is likely to pay off very quickly. When taking this approach, it may be important to track the progress of each programmer, especially in a large department (for instance, by keeping track of which CRF domain and/or process part each programmer has worked on).

6. USING YOUR ACQUIRED SKILLS

Indeed, following on from the previous section, variety is the key to continuous professional development as programmers become more senior. At least, within reason. To some extent, it will be important to consolidate newly acquired skills through repeated use (as mentioned in section 3), however exposure to a variety of new scenarios will expand career development possibilities. As before, this applies equally well to all types of skills: technical, procedural, therapeutic, team skills, management skills etc.

Within projects, programmers will want to gain experience of all parts of the process (e.g. programming plans, derived dataset programming) and all aspects of clinical trials (i.e. different CRF domains, CDISC etc). However, it will be part of the manager's role to ensure that the more experienced programmers contribute far more to the organization than just their project programming work. For example, more experienced programmers may wish to contribute to organization-wide programming initiatives (e.g. development of in-house macros, or in-house training).

Almost inevitably, as programmers become more experienced, they tend also to become more valuable (and desired!) team members. As a consequence, they may find themselves being assigned to more important projects (or more important roles on their current project). This, of course, is a good sign! But there are some potential snakes to be wary of. While Project Managers (and statisticians!) may be used to statisticians leading teams, they may be less familiar with statistical programmers undertaking this role, therefore programmers may have to go the extra mile on their first such project in order to prove their worth. Team leadership requires significant soft skills (communication, motivating staff, team skills, time management, negotiation etc) and the challenge for programmers undertaking such a role for the first time should not be underestimated. Certainly, some form of training will be beneficial, as would the existence of reference documentation (e.g. Project Start-up Checklist, Team Leader Role and Responsibilities). It may also be worth considering assigning an experienced Team Leader as a routine team member to act as a mentor for the Programmer in their first team leadership role.

Yet again, the win/win principle is not far away. The more skilled your programmer becomes, the more the organization can gain from their knowledge. And, of course, the more their own career options open up.

7. PLOTTING A PATH TO SQUARE 100

When considering career development options, it is important to consider long-term ambitions as well as short-term goals. The nearest step on the ladder may seem the easiest one to reach, however it may lead to an immediate snake to take you away from your chosen path. Perhaps surprisingly though, snakes are not always bad. It is often said that anyone can do a job if it's easy, but the real test of a leader is how they fare when they're faced with a difficult challenge. The wily professional may deliberately look for snakes to illustrate how adept they are at bouncing back.

Your statistical programmer has now acquired and practiced the basic skills extremely effectively and picked up a few more skills besides. They've proven themselves to be technically sound and excellent leaders. So, where do they go from here and how do they get there? As a manager, your role will be to facilitate your programmer's advancement

PhUSE 2008

as far as their potential will take them. If this is high up in an organization, then early exposure to senior managers may well be beneficial. This could be anything from representing you in a management meeting (e.g. if you are away) to representing the department in a global initiative. Equally, to enable a broader understanding of the industry as a whole, you may want to encourage your programmer to get involved in a scientific organization, perhaps even presenting at a Conference...

With good grace and a following wind, your programmer will finally reach their desired career destination. But by then, their knowledge and experience will have broadened markedly and, not surprisingly, they will probably have developed new career aspirations. The game of snakes and ladders may be over, but a new game is just about to begin ... Battleships perhaps?

CONCLUSION

In this paper, we have considered the art of career development for statistical programmers. The importance of equality between statistician and programmer career paths has been highlighted, and the role of the organization and the employee in this regard has been discussed. The need for focused training and development at both the induction and continuous professional development level has been stressed, with an emphasis on the hands-on approach. Ultimately, the win/win principle must apply – if the organization does its part (structured career ladders, manager-led career development etc) and the programmer does their part (proactive learning, effective programming and strong leadership), the game will end with everyone a winner.

REFERENCES

1. Briggs Myers, I; McCaulley, MH; Quenk, NL; Hammer, AL. *MBTI Manual (A guide to the development and use of the Myers Briggs type indicator)*. Consulting Psychologists Press, CA, 1997
2. Kolb, DA. *Experiential learning: experience as the source of learning and development*. Prentice-Hall: NJ, 1984.
3. Gordon, K. Who wants to be an ICH euro billionaire?, *Pharmaceutical Statistics*, 2008, 7, p4-8.
4. Goleman, D. *Emotional intelligence*. Bantam Books: New York, 1995
5. Learning and Development Roundtable: <https://www.ldr.executiveboard.com/Public/Default.aspx>

ACKNOWLEDGMENTS

I would like to acknowledge my Quintiles colleagues, Leo Valmas for reviewing and improving earlier drafts of this manuscript and Jo Briggs for her valuable suggestions.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Dr Kerry Gordon
Senior Director, Biostatistics
Quintiles Limited
Station House
Market Street
Bracknell
RG12 1HX
Work Phone: +44 (0)1344 708281
Fax: +44 (0)1344 813281
E-mail: kerry.gordon@quintiles.com
Web: <http://www.quintiles.com>

Brand and product names are trademarks of their respective companies.