BEST practices

A Publication of
The Center for Information-Development Management

MARCH 2022
VOLUME 24, ISSUE 1

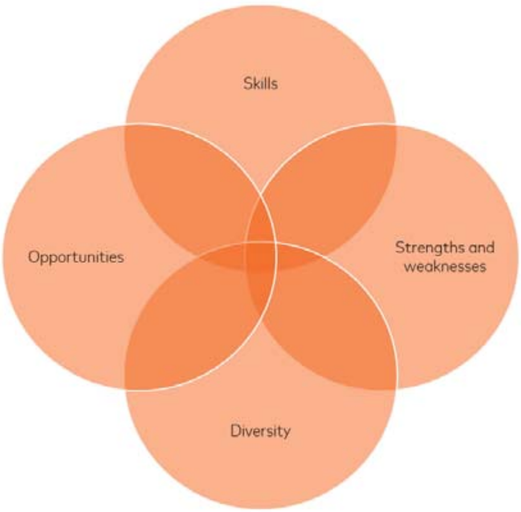
Creating Cohesive Writing Teams

Tricia Kinman and Rich Gebhardt, Mastercard

As the job description for technical writers has increased and expanded, the need for managers to create team cohesion is not only significant for the everyday work environment, but it plays a vital role in individual development.

Traditionally, technical writing has been a siloed enterprise, with writers working with individual subject matter experts in standalone software. With the movement from standalone environments to content management systems, there is an increase in the need for collaboration and shared responsibilities. Today, there is more of an emphasis for technical writers to be a team player, work cross-functionally, and to have a consistent approach for all writers who engage with outside business partners. For managers, team cohesion can have the practical advantage of helping train and properly staff for high volume periods.

Figure 1: Venn Diagram - Core Attributes



Hiring and Assembling Teams

So how do you create cohesive teams? It begins by hiring and assembling teams with thought and precision. I like to think of a team as a Venn diagram, with all having a core set of attributes yet each person contributing something unique. When hiring, I look in 4 areas:

Skills

What skills does the team currently lack, and does the candidate complement the team by bringing a valued skillset to the table?

Strengths and Weaknesses

What are the candidates' strengths and weaknesses? By pairing this candidate with others on the team, will I enhance their natural talents and have a way to balance out their weaknesses? This creates a situation where they need to rely on their team members for support.

Diversity

Does the candidate bring fresh perspective and diversity to the team? Diversity on a team promotes alternative views that challenge the status quo and help implement change. A team of all the same people with the same backgrounds and attitudes doesn't lend itself to innovation. Also, our customers and subject matter experts come from varied backgrounds, and the team should mirror this dynamic.

Opportunity

What opportunities can I offer as a manager for growth and challenges? Will the writer have the ability to learn and obtain a new skill set or enhance their career? Having opportunities to learn and progress empower individuals and give them a "place" on the team.

(continued on page 5)

CONTENTS

Best Practices Newsletter

A publication of The Center for Information-Development Management.
710 Kipling Street, Suite 400
Denver, CO 80215
Phone: 303-232-7586
Fax: 303-232-0659
www.infomanagementcenter.com
info@infomanagementcenter.com

Publisher and CIDM Director
Dawn Stevens

Production Coordinator
Lisa Lambert

How to submit an article:
Kathy Madison
kathy.madison@comtech-serv.com

How to join the CIDM:
info@infomanagementcenter.com

From the Director

Dawn Stevens

Lessons from Daddy

Dear Readers,

On February 3, 2022, my father passed away unexpectedly at the young age of 77. I've always been the quintessential daddy's girl, sharing that special bond only known between father and daughter, and I will miss him terribly. Through our relationship, he had a profound impact on who I am, how I work, and how I run my business. He assured me I could do whatever I put my mind to, he taught me to work hard and play hard, and he encouraged me to be a person of integrity. In tribute to him, I want to share a small selection of important life lessons learned at his knee from his loving example.

Life's Not Fair

I can't even begin to count the number of times the words "That's not fair" came out of my mouth in conversations with my dad. As any child would be, I was filled with righteous indignation at the injustices that I encountered, often at the hand of my father – my very own flesh and blood! I was being held to a higher standard or a different rule set than my friends. My little brother was being given privileges that I did not have at his age. I couldn't do whatever it was I wanted because we couldn't afford it or because I had responsibilities at home.

Clearly, in some cases, I was in the wrong, measuring fairness by a childish and selfish perception of right and wrong. In many other cases, I was right. No, it wasn't fair that my mother was very ill and that, as a result, I had to grow up a bit faster than some children, taking on extra responsibilities for household chores and watching over my brother. No, it wasn't fair that my dad was laid off, so we had to move to a new state or we didn't have the money for some kind of special toy or event. I'm sure he often thought the same thing—it's not fair—but regardless of the situation or the correctness of my statement, inevitably, his response back was always the same: "Life isn't fair."

Fortunately, he didn't just teach me this trite phrase, but also how to deal with the inherent inequities of life. Through his example over the years, I learned that while life wasn't fair, I still had control over my choices for how to deal with that fact:

- I could continue to rally against the inequity, whining and moaning, portraying myself as a victim, and telling all my friends and family how rotten and unfair my dad was. This route never accomplished a single thing. Sure my friends might sympathize and agree with me, but even if they had stood in solidarity beside me in front of my father (no one ever did), they had no power to change the situation.
- I could resign myself to the situation, taking on the role of martyr, quietly suffering for adolescents everywhere. Although this route could occasionally make me feel better, (knowing that I was taking one for the "team"), it also got me nowhere. My martyrdom did not inspire a revolution to avenge my treatment; no hunger strikes to bring about changes in the parenting world.
- I could work to understand the circumstances that caused the inequity, learning to see things from another angle, from another's perspective. Often understanding led to acceptance, an empathy for the inequity that the other person might also be experiencing. Sometimes it led to a clear path for rectifying the situation to be fairer in the future or avoiding the situation in the first place. I learned not to blindly cling to my beliefs, but work to understand all sides. It was only then that I might see a path to make things better or to make the most of a bad (read, unfair) situation.

It would be a lie to say I took the high road in all situations. It was (and still sometimes is) a hard lesson to learn, and human instinct is to demand justice and fairness. However, I can look back and easily see that my dad wasn't always given the recognition or rewards he deserved, and he often got the short end of the stick; nevertheless, he taught me to play the cards I was dealt.

That lesson was taught both figuratively and literally. My dad loved to play games and he taught me several trick-taking card games at a young age. My favorite was three-handed cut-throat pinochle. I learned to turn bad hands to my advantage, to strategize and bid appropriately. Every hand wasn't fair; sometimes, you were dealt lots of aces and you could coast your way to victory. Sometimes your hand stunk, but if you worked at it, you could still make forward progress toward the end goal of winning. Cards aren't fair, but you can learn to play them anyway. Life isn't fair, but you can learn to live it anyway.

The lesson certainly carries over to my career and my business. Sometimes, things don't go the way I expect or hope. A deadline requires me to work a weekend or a holiday. A project scope unexpectedly expands or collapses. An employee requires a long-term absence. A pandemic cancels all in-person conferences. It's not fair. But I remain in business because my dad taught me not to dwell on the unfairness of it all, wallowing in self-pity, but to change my perspective: to make the most of the situation, do the best with what I have, not to be afraid to try something new, and, of course, if I don't succeed with my first approach, to try again.

Take it or Leave it

As I learned perspective and how I might try to transform situations that I perceived as unfair into tenable solutions, I encountered my next lesson. Negotiation wasn't always an option, as sometimes my dad would dig in his heels and let me know my only options were to "take it or leave it."

Early encounters with the phrase "Take it or leave it" were almost always in the context of my dinner. I am now and have always been a picky eater. Honestly, my parents tried a number of methods before resorting to this phrase, including arranging the food on my plate into a smiling clown face and later, putting me in the chair that faced the outside window where I could see them playing catch while I was left to finish my dinner. As I grew older, however, they worried less about whether I would be malnourished and adopted the "take it or leave it" approach. I didn't have to eat what I was given, but I wasn't being given anything else.

The take it or leave it mantra showed itself in other situations as I grew up. You can invite this many friends or none at all. I'll buy you these select (non-designer) clothes or you'll get nothing new for the school year. You'll be home by ten or you won't go. In virtually all cases, the choice was a no-brainer – something was better than nothing (although there were definitely situations where not eating was preferable to what was being served).

The lesson has mixed results in my life. I am a horrible negotiator at garage sales or with street vendors in foreign cities. If I want the item, I will buy it at the listed price because I assume the price is offered with a take-it-or-leave-it attitude. Similarly, it annoys me when someone offers me less than what I asked for on Facebook Marketplace. If you don't want it, fine, but I already told you what it would require for you to take it.

If you Can't Take it, Don't Dish it Out

My dad also taught me to avoid problems by not causing them in the first place. My actions sometimes prompted situations that made me unhappy, and in fact, often led to my complaining about the unfairness of it all. In addition to learning that life wasn't fair, my dad also gave sage advice that what goes around comes around, and if you can't take it, don't dish it out.

As I mentioned, my dad loved to play games. I loved to play them with him and I loved (well, still love) to win. My aggressive strategies, however, were not always appreciated by my opponents. For example, we played Risk a lot, and I would gleefully work at amassing my armies to take over one opponent at a time, picking on the underdog in some manner to further my cause. How I hated the resulting reaction from the rest of the players in the game. Suddenly, all eyes were on me and all opponents worked in conjunction toward my eradication from the game. I spent many a game in tears because everyone was against me. No sympathy came from my father, just the statement, "if you can't take it, don't dish it out."

I am four years older than my only brother. For the entire time that I lived full-time in the same house as him, I was taller and heavier than him. It was safe to pick on him – I could pin him to the ground easily if he came after me for my teasing, my smirky faces, or my pushing, punching, or pinching when parents weren't looking. My parents weren't blind to the situation, and I certainly got in trouble for my actions, but in conjunction, I was also warned repeatedly, "if you can't take it, don't dish it out." These were nonsense words to someone of my infinite wisdom. Take what? Yes, I knew my brother would eventually get bigger and stronger than me. I also figured that would happen after I went to college. I was right on both counts, but failed to think about my breaks at home between semesters, when my fully grown little brother would gain his revenge. During that time, little punishment was actually doled out. My brother received no words of wisdom; instead, my dad would smile at my pleas for help as I was pinned to the ground and simply say, "I warned you."

When you think about it, the expression, "if you can't take it, don't dish it out," is a corollary to the golden rule – treat others as you would like to be treated. If you don't want to be picked on, don't pick on others. If you want respect, give it. It's an important lesson for playing games, but even more important for running my business. Would it be easier without competitors? Of course. I could take the stance to ban such competitors from my conferences, refuse to let them advertise or speak. But how does that serve the community and what does that do to my reputation? If I can't take the disrespect those actions would afford me, I better not dish it out. Instead, I can recognize that there's room for us all. My competitors actively participate at my conferences. They invite me to speak with them at other events or to have a chat on a podcast. We refer clients to each other based on the strengths we each offer. In direct competition, sometimes I'll win a bid; sometimes I'll lose. But in the end, we all gain – the industry is better as we all work together and in competition to improve technical communication practices.

My dad's passing reminds me that we only have a limited time on this earth, and our best hope is to leave a positive legacy before we go. It is my prayer that the lessons I've shared here are apparent in all I do, and that I am a living tribute to the man my father was.

Dawn

(continued from page 1)

Thoughtful Task Assignment

Once the team is assembled, managers can create cohesion through the thoughtful and purposeful assignment of tasks and responsibilities. As a manager, I like to empower each team member to own a segment of assignments or a product. Then, I create processes and specific tasks that touch all assignments so team members must rely on each other. For example, everyone might be responsible for individual documents for a given product, but quality and peer reviews are a shared responsibility, making the entire team responsible for accuracy of deliverables.

Another way to create team cohesion is to have monthly or annual sessions for writers to collaborate on best practices and give input into documentation design. If someone has a new document, a brainstorming session to discuss design strategies together can promote collaboration.

As teams work together to collaborate and meet deadlines, assignments can be modified as needed to suit priorities. By giving everyone shared responsibility for the content, writers can easily step in as needed and get support.

Collaboration is a Team Sport

If working together is something we were taught when young….why can it be so challenging in the business world? Like many of you, our technical writing teams here at Mastercard work with others across the globe. The concept of cross-functional remote teams is not new, but it still can present challenges. Often we have to balance different needs:

- Writer's needs: what is the information I need, who are the key stakeholders to confirm/approve, are they responsive, are we aligned with the stakeholders so we meet all established deadlines
- Stakeholder's needs: features/stories thoroughly developed, what development needs to occur, what release is this tied to, is there a change in scope

The team I currently manage is embedded – so writers work closely with key stakeholders – developers, product owners, SMEs and attend scrum and project planning meetings to understand what is needed. it's like putting together pieces with many different hands, to make a final product – like Legos…or like cooking…combining all different "ingredients" – the information we obtain, different stakeholders we work with – all into one "recipe" which is the final product that we put out.

Tricia Kinman
Mastercard

Tricia Kinman manages an embedded writing team for Mastercard, a global financial service corporation. Her team produces technical, user, and API documentation for fintech customers. She has been working in the technical writing industry for 10 years as a technical writer, information architect, and manager. She is passionate about problem-solving, designing solutions to

Rich Gebhardt
Mastercard

Rich Gebhardt is Manager of Content Strategy and Development at Mastercard. He has 17 years of experience in the industry as a senior technical writer, consultant, and manager. His team is comprised of embedded writers, working very closely with products owners and developers to create content that supports Mastercard products and services. He is passionate about how the content we provide impacts the customer experience.

Writers wear many hats – they may not be the SMEs or product owners – but they should be the experts in **what** we need to provide to customers. Writers have to:

- Leverage tactics to help decipher the key information that needs to be announced
- At different times, they have to rely on skills that help promote collaboration, such as research/interview skills, project management, or their leadership – to help keep projects on track or raise awareness to potential gaps

As a manager, part of my job is to use collaboration to help promote success – not just for our team but for the scrum teams, SMEs, and other stakeholders we rely on. As part of a team, I want to:

- Help the team work together more effectively – through collaboration comes greater value to what we can provide
- Help them focus that what they're doing is important to support the customer experience
- Remind them that their goal is to not just help improve their work – but to improve the **impact** of their work

As a manager, it's important to remember that you will not always have the answer…or even the right answer, so it's important to be an active listener – sometimes you need to listen to find the next right question. In problem-solving – rather than give step-by-step instructions on what to do – offer empowerment and give ownership to help them find solutions to a question that they may have or to a roadblock that stands in the way of what they think is needed. This helps build expertise, knowledge, and drives a sense of inclusion. It also helps ignite change to unlock potential and expands the possibilities of the entire team.

Collaboration Offers Room to Grow

Any roadblocks you can't work through could result in unclear or incomplete content because you were not cohesive as a cross-functional team. It also impacts customer inquiries received, as incorrect content requires us to determine the issue and then revise. Each negative experience resonates to customers and impacts their trust – think about a negative customer experience you have had. It's like building a house – it takes time, months to build…but can be damaged in just a short amount of time, then takes time to build back up again.

By keeping a sense of inclusion, encouraging diverse perspectives, and focusing on the end goal (in our case, the customer experience) if these concepts are put into practice every day, then the result is improved clarity and consistency for our customers. That in turn, makes a team, department, group, and company you are a part of – not just look good, but also shows that you have the customer experience in mind.

To provide clear and accurate content, we should be writing from the "outside in." What does this mean? It means we should look at what we're working on and think about a product or service from a customer's perspective. They are relying on the content to help get their job done and to help them accomplish a goal. How do we know if we are missing a mark with customers? If you hear:

- I can't find what I need
- What does this mean?
- How does this impact me?
- Why should I care?

Then you know somewhere you have missed the mark. We're all customers of something right? As a customer, what do you want to know?

- What is this product?
- How does it work?
- What do I have to do?
- How does it benefit me?

When you think about **who** your user is, your writing can target **what** your user needs. Understanding a customer's needs or goals and thinking from their perspective—from the outside in—helps us provide the right level of detail and appropriate **context**. Content that does not have details about a product/service and how to use it, doesn't let them know its value or how it helps get their job done. *When content is more relevant to a customer…our value is more relevant to the process.*

Adding Value

Clear, accurate content – and writers – add value to products and services. Research shows that tech customers spend 55% of time with the content vs. 21% of time with sales teams. IBM has shared that:

- Tech content of high quality had a 84.9% impact on the **perception of the company**
- Tech content had 96.3% impact on **perception of a product**
- 92% impact on the **satisfaction of the product**

Something else to consider when working with teams is to trust yourself. Always ask (and trust)…that if something does not make sense to you, chances are it will not make sense to a customer.

As we partner with internal teams that may be so focused on the business need, it's our job to remember that collaboration and consistent communication not only builds up teams, but it adds to the value a team can provide, and results in good content. What can you do today to help shape or influence the information that you provide to your customers tomorrow?

Goals and Outcomes

The goal of creating a cohesive team is to establish trust, create shared experiences, and develop a partnership in seeing each other succeed. The outcome, if the team is cohesive, is an improved work experience which ultimately contributes to better content for customers to consume.

Not THAT Self-Service

Lawrence Orin, Bringg

Lawrence Orin
Bringg

Lawrence Orin is a thought leader in technical product content, and has led documentation teams for much of the past 20 years. He has advised numerous companies on their content strategy, and currently leads the Product Enablement team at Bringg. Beforehand he was offering content solutions at Zoomin, and led doc teams at Radvision/Avaya and Aternity/Riverbed. He has also led teams in customer service and tech support. He has a background in programming, graphic design and content creation, and holds a B.Sc. in Computer Science with Cognitive Science from University College London.

I recently started working at a company to create their content vision and build out their documentation and training. It's part of a company-wide initiative to promote customer self-service, by offering self-help documentation and self-configuration, where users can access the settings they need to make their own changes.

Self-service as a strategy is just common sense. When customers can configure the product independently, without requiring them to contact customer services to ask for customizations, everyone wins. The company saves costs on support and implementation, and customers can use the product to its full potential.

It's what we've come to call the enablement equation:

> **Enablement = My Configuration and Content**
> Or
> E=mc2

While it's obvious that self-service empowers our customers, it's also worth remembering that we, as content creators, are also customers, where our tools are for authoring, publishing, managing content, and more.

Like many writing teams, we sketched out the user experience we wanted for our audience, then drilled down from the broad vision into the details, eventually arriving at our list of requirements for each tool we intended to buy.

Then, armed with our list of requirements, it was time for shopping! We joined dozens of sales calls from a bewildering array of vendors, all trying to convince us that their product was the perfect match for our requirements.

All the while, we quietly held on to the belief that our tools must respect the enablement equation, E=mc2, to allow us to self-configure, to empower us to achieve our own goals whenever we wanted.

> ***"Every tool should empower customers with self-service"***

Fast-forwarding through all the sales calls, comparing each requirement to the features of the product, we ended with:

- A flat-out "no" if they missed any core requirements, or
- An obvious "yes" if they met our requirements, or
- An uneasy "hmmm" if they could potentially meet some requirements because it's "in our roadmap" or they just need to do "a short customization".

Whenever we heard "it's in our roadmap" or "sure, you just open a ticket and we'll customize it for you", it made us feel uneasy.
In a way, it sounded ideal, they would tailor their product to create an exact fit for our requirements, right? So why were we uneasy?

"It's out of the box!"

All tools allow their users to customize features. Anyone can customize a word processor's display of *Heading 1* to be in Arial 18pt bold. Everyone who owns a smartphone can customize it to display a custom background image.

That's an out of the box *configuration*, not a *customization*.

So, when does a *configuration* become a *customization*?

Configurations are available for customers to independently change the behavior of a feature. When it's a fully supported setting, users can feel safe knowing that as the tool grows and evolves over time with new releases, the setting will continue to be supported in future versions.

Internal customizations, however, are very different. They are not available to customers, because they require deeper changes to the tool functionality.

> ***"When does a configuration become a customization?"***

If a salesperson says that a platform can perform some functionality, but it must be customized, try reacting in the same way any good writer would react: turn that passive sentence (*'must be customized'*) into an active one. Ask the key question, "*Who makes the customization? Can I do it myself? Or am I dependent on you?*"

The Balance of Power

When a vendor customizes on your behalf, it's both easier and harder. It's easier because you just sit back and let someone else do it for you. But it's also harder because you are at the mercy of someone else's priority list, their available resources, their sense of urgency, and their timetable.

In some cases, you open a ticket, you wait, you follow up in your weekly meetings with them, they promise to get back to you, you follow up again, eventually it gets scheduled, you follow up again, perhaps it gets delayed, then finally they deliver the functionality, you check it works, you uncover problems, you open more tickets, and so the joyous customization wheel keeps turning.

If you're not sure who does the customization, ask (even demand) to see the configuration page. Take the time to explore the settings yourself, to discover what you can control versus what's under the vendor's control.

This determines whether it's a simple setting (configuration), or if the vendor must customize it for you.

> ***"All enterprise platforms need to be customized, don't they?"***

The Customization Merry-Go-Round

Customizations have another danger, especially with SaaS systems. As the core platform evolves, there is a chance that your custom changes might break with the new version. This would not happen if it was the result of properly supported customer-facing settings. But when they add custom code just for you, it's not at all clear if there is a commitment to support this behavior in perpetuity.

Once again, you are at the mercy of the vendor's priorities. You must either delay your upgrade, or you fasten your seat belt and embark on another ride of the customization wheel of joy, with a new set of tickets, a new round of waiting, scheduling, confirming, discovering bugs, more tickets, and generally more joyous joy.

Before the advent of SaaS, most enterprise platforms required an integration phase, to connect with existing systems and configure the behavior you need. This led to the popular adage that "*All enterprise platforms need to be customized.*" There's that pesky passive sentence again. Who does the customization?

"Ask not what your vendor can do for you. Ask what you can do for yourself!"

That may have been the paradigm 20 years ago. But today, modern SaaS enterprise systems like Google Suite, Paligo, JIRA, Monday.com and dozens more, all work with large companies and all respect the enablement equation: they offer self-service to empower their customers.

Consultants are not part of the hidden customization world. They are part of the self-service paradigm and therefore they respect the enablement equation. When you employ consultants, you pay experts for their knowledge and experience to do the work for you. They use their know-how to configure customer-facing self-service settings. Consultants serve you, not the other way round. They are dedicated to your timetable and your priorities. That's a world of difference from asking a vendor to give the functionality you need from settings which are hidden from users.

Customize with Templates

Some vendors try a different approach to customizations; they create groups of predefined settings. Each *profile* or *template* can hide large numbers of complicated fields under a single umbrella.

You can see this approach in applications which offer preconfigured permissions for standard user profiles like an admin and a viewer.

You may have seen umbrellas like this when you set up a new email on a phone. Instead of wading through settings for mail synchronization, port numbers, and server addresses, most people choose the mail settings by picking from a short list of popular providers, which preconfigure those settings behind the scenes. A user only needs to enter an email address and password, and they're good to go!

"With templates, it's all or nothing"

However, these profile examples can be tricky, because if you find yourself outside all the preconfigured mail settings, you can still self-serve by configuring your own email provider, by filling in some user-facing settings.

It brings us back to the same question which still needs asking: *"If I don't fit one of your templates, can I self-serve, or must I depend on you?"*

For example, when you switch on low power mode on your phone, it masks hundreds of small optimizations to use less power. If you're lucky, you might catch 10% of them in the UI, but most are simply unavailable to customers.

In the product content world, you can see this in some documentation portals which offer customers a set of templated website layouts for a quick setup. They have a fixed look with only very narrow variations, because they hide often thousands of internal-only settings. If your vision doesn't fit a template, you're either left on the sidelines, or you're invited to join the customization merry-go-round.

A fair solution which empowers content creators would offer a tool where you can design your docs portal independently, like Wix or similar web design tools.

Customize with Apps

Another way for vendors to allow their customers to self-serve can be to turn their product into a platform and let customers develop the code for their own functionality in the form of plugins or apps. DITA, the open standard for technical documentation, is an old example of an architecture which uses the plugin philosophy extensively.

If you have coding skills, plugins can be a neat way to independently add your own functionality to fit your requirements, and of course the vendor wins by offering a safe and scalable way to extend the features and value of their platform.

Plugins morphed into apps, as the world moved online to SaaS. It became possible to add an open marketplace for developers to upload their plugins and offer it to others. Pioneers in this space were web platforms like WordPress.

Letting customers self-serve was a massive boost to their popularity. WordPress became an overnight sensation, as customers could add any third-party plugin they wanted, or create their own. The platform could suddenly do so much more.

But in those early days, people quickly found that as the underlying platform was upgraded, the plugins also needed to upgrade with it, which became a problem when some developers didn't upgrade quickly. Anyone dependent on a plugin for added functionality suddenly found themselves back in a position of powerlessness, waiting for someone else's timetable. Another common complaint was that many plugins were incompatible with one another, as they clashed over the same resources and features.

> ***"An app developer can self-serve with custom functionality, but you need to know how to code"***

This led to the invention of the *curated* App Store, popularized by companies like Apple and later Google, Microsoft, Salesforce and many others. In a curated store, developers must submit their apps through a strict approval process, to ensure their code obeys the platform's technical rules, like remaining within their allotted memory space to avoid compatibility issues, or that they only use the platform's official APIs, so they remain robust over time as the underlying platform evolves.

The plugin/app paradigm demonstrates how self-service can be wildly popular, and offers the classic win-win for vendors and customers. The only downside is that you must know how to code, and often it's a proprietary language. For customers of third-party plugins, they put you back in a position of relative powerlessness, as you wait for someone else to address your needs.

Self-service: the sweet spot

Whenever a tool offers the ability to make their own customizations as self-service, it's usually a huge win. If customers are forced to ask a vendor to customize the tool to get the required functionality, it's a road to anguish on all sides.

So here are some pointers for your next shopping trip.

Beware:

- If you hear a tool has some "out of the box" capabilities, play with the settings independently. Check you can access the settings you need to meet your requirements.
- If a tool offers templates as an umbrella to a multitude of hidden settings, make sure the "out of the box" templates fit all your requirements, without the need for further customization.
- If a vendor offers you to add your own custom code to achieve the functionality you need, make sure the language is well known (like JavaScript, CSS), so you can easily create it or find a developer who can do it for you.

Ideally:

- Have a clear content vision and distill it to detailed requirements.
- Verify a tool fulfills all your requirements.
- If you need to add behavior, check you have the flexibility to customize independently to achieve your requirements.

The Content Scientist

Michael Iantosca, Avalara, Inc.

Michael Iantosca
Avalara Inc.

Michael Iantosca is the Senior Director of Content Platforms at Avalara Inc. Michael spent 38 of his 40 years at IBM as a content pioneer – leading the design and development of advanced content management systems and technology that began at the very dawn of the structured content revolution in the early 80s. Dual trained as a content professional and systems engineer, he led the charge building some of the earliest content platforms based on structured content. If Michael hadn't prevailed in a pitched internal battle to develop an XML platform over a planned SGML variant called WebDoc in the mid-90s at Big Blue, DITA, and the entire industry that supports it, might not presently exist. He was responsible for forming the XML team and a member of the workgroup at IBM that developed DITA.

People that know me know that I am rarely at a loss for words. Like many of you, words are my life's work. The only time I get tongue-tied is when my wife asks, "*What should I tell people that you do for a living?*" Uhhh…

that question was easy to answer early in my career. I could simply respond, *I'm a technical writer; I write instruction manuals* or respond with any of several discreet roles in the field we refer to as Information Development or User Experience. It's not so simple today. Many of us concurrently serve various roles among a diverse array of skills that we've gained almost exclusively on the job.

Labeling my career rarely interested me until an executive residing over an Information Development organization billed themselves as a Content Scientist. That got me thinking, what qualifies someone as a content scientist?

Many of us are, in fact, Content Scientists. We have evolved to that role as we've acquired a broad range of interdisciplinary skills learned on the job, often over many years. Moreover, as our discipline has evolved, the array of essential skills has grown dramatically as we've advanced the discipline. Once upon a time, we hired graduates from a wide variety of college majors such as history, English, and political science, then trained them as newly minted Information Developers. Technical writing degree programs were rare; most gained their skills on the job.

That's no longer tenable; we need candidates with key skills and foundations coming out of our universities. We need to transform university degree programs in content from the arts to the sciences–from BA degree programs to BS degree programs and develop Content Scientists. We need to elevate our profession as it has become a profession of the sciences in addition to language and communication arts.

The root of today's skill deficit coming out of universities stems from the evolution of what were and remain separate degree programs. Taken individually, the resulting skillsets have become insufficient for our field. By luck or circumstance, I created an interdisciplinary program for myself that crossed multiple degree programs spanning communications, computer science, and psychology while underpinned with a strong core liberal arts curriculum. At the time, I didn't know anyone in my undergraduate or graduate programs that crossed these program lines in the late '70s. Little did I know how well my computer science coursework would serve me when I was hired by IBM as a Junior Information Developer. New Information Developers endured intense programmer training programs in months-long classes side-by-side computer science valedictorians from elite universities. Those folks read thousand-page computer dumps in hexadecimal as if it were the Sunday paper. Failure to make a passing grade on grueling coding exams for weeks on end resulted in immediate dismissal–for a job as a technical writer!

When I joined IBM's Information Development, the company had an extensive in-house curriculum and instructional staff that taught week-long classes for Information Developers. The company would regularly fly us, carte blanche, to in-person training venues. Those days are long gone. Such in-house training was rare then; it's nonexistent today.

"*OK, Boomer,*" you might rightly accuse, but I digress.

What coursework would a premier undergraduate and graduate program in Content Science consist of? This is a question I've been pondering ever since I first heard the term uttered. Your list may differ, but I'll float a proposed starter set here to seed the discussion.

Before we continue, allow me to make two important disclaimers. First, an undergraduate program need not dive excessively deep into every subject area; that's the role of graduate and doctorate programs. Nevertheless, such a program needs to provide practical hands-on experience and be interdisciplinary – combining what has traditionally been separate programs, especially those that span Communications, Engineering, and Library Sciences – underpinned with a strong core curriculum.

Second, I am not suggesting that such a curriculum should exist to simply mint fresh Content Scientists only to serve the needs of business. That's not the role of the university, but the success of its graduates is nevertheless an important measure of any program. It is equally about preparing new graduates for long, diverse, and prosperous careers with a high degree of mobility, professional recognition, respect, and value that is often lacking in the perceptions of peers in engineering and by corporate leadership. We must define our profession and the unique and valuable skills we bring to the business instead of allowing others to continually define us. As the saying goes, perception is reality; we need to manage how others perceive us.

> ***"It's telling how quickly AI technology such as DeepMind is advancing.***
>
> ***DeepMind's new AI coding engine is as good as an average human programmer (The Verge, 2/2/22). While robotic content also continues to evolve, neither can replace human declared intent. There will come a time when we'll write intent-based content that drives robotic software generation. It's already happening with robotic software video generation with systems such as Videate™ and robotic writing applications such as Jasper™.***

Core Curriculum

Before we delve into the major itself, a foundational core curriculum that spans the liberal arts and sciences is essential. Such a core creates a basis for basic skills, business acumen, career mobility, and connects-the-dots beyond domain specialization. I'd advocate for a basic core that includes rhetoric, philosophy, psychology, sociology, business, math, basic engineering/programming, and more. Some schools are adamant and mandate a core curriculum. I am certain that the core curriculum that was required by my college program at Marist College served me extremely well; It was key to my career ascension and path through virtually every role in this business we call Information Development.

Undergraduate Major in Content Science

Let's explore potential foundational coursework and elements that might establish a credible university degree program in Content Science.

Foundations in TechComm and MarComm

There's a core set of courses and subject matter that a robust program in Content Science ought to include, in my opinion. This is not an exhaustive list but a relevant subset that I'd strongly recommend. When I'm interviewing candidates, those that have a robust cross-section in these areas are the ones that impress me most.

TechComm and MarComm 101

This is the typical course one might find at many universities. Even when a school has a degree program in Communications, Computer Science, and Library Sciences, there is little consistency under which school such programs reside. Ironically, an inquiry I made to the School of Communications professor at one university directed me to the School of Computer Science, while another directed me to the School of Communications, and a third to their School of Library Science. It's a living and breathing example of "*Who's on first base, What's on second base, and I Don't Know – third base.*"

Of course, we want such courses to cover the basics: core writing skills, editing for quality, and so on. However, there doesn't appear to be consistency in other areas such as progressive disclosure, minimalism, componentization, structured vs. unstructured authoring, designing, writing for reuse, and more. With MarComm, there's a strong intersection with persuasion, advertising, and marketing communications courses often included in Communications curriculums and psychology courses (think Maslow's Hierarchy, the Johari Window, and so on).

Foundations of Content Science

The next set of courseware I'd suggest provides a broad base of foundational knowledge across an array of core content practices. Some are extensive and deep enough to warrant full-semester courses, while others might be substantial subsections of combined coursework that go beyond skimming the conceptual surface and build working knowledge and proficiency.

- Information architecture, including content models, content types, content access models, content use models, and macro/micro-content journey mapping
- User experience design foundations and embedded user assistance models and methods
- Survey of authoring models and tools including structured, semi-structured, unstructured, single-sourcing, reuse and repurposing, micro-content architectures, document object models, inheritance, and specialization.
- Visual communications, including scripting, manual and robotic tour generation, graphics, and graphics management
- Content analytics, measurement, and effectiveness
- Fundamentals of content personalization
- Foundations in globalization, localization, and internationalization
- Applied computational linguistics for content governance
- Foundations in terminology and terminology management
- Foundations of artificial intelligence and machine learning for content
- Foundations in semantic science, including taxonomy, ontology, and knowledge graphs
- Survey of content supply chains and content supply chain models
- Fundamentals of knowledge-centered services (KCS)
- Design systems
- Topics in software development lifecycle (SDLC), agile methodologies, continuous delivery, and DevOps
- Content operations management, including financial, planning, processes, and governance

As I suggested earlier, this is only a seed list. I am confident we could more than double or triple it in a shared whiteboard setting.

Such a program might exist, but I'd suggest that would be more the exception and not the rule. I have been surprised (though rarely) to encounter a candidate that not only has a degree in TechComm but could also converse fluently about ontologies and knowledge graphs. As I draft this article, my organization is actively seeking a hybrid terminologist/taxonomist with those basic skills and at least a keen interest in ontology, graphs, and artificial intelligence. It would be a breath of fresh air to meet university graduates that could not only hit the ground running with those skills but graduates that possess a well-rounded foundation in areas such as Omni-source and omnichannel content models.

Post-Graduate studies in Content Science

We also need to help develop post-graduate studies in content science to enable depth and discovery for our discipline. This is where university partnerships become invaluable and can be of mutual benefit. We need to take the most vexing content challenges we face and leverage such programs.

How often have we asserted that post-sales content contributes to revenue, only to duck when challenged for empirical evidence to prove, let alone track it? How often have we said we're seeking to apply semantic intelligence with artificial intelligence and machine learning to enable dynamic and personalized content retrieval, organization, and delivery–even autonomic content? Or how we might apply emerging technologies such as blockchain and NFTs that can enable monetization of a semantic web of content shared across multiple providers and copyright owners? These and other technical and content management frontiers are ripe for post-graduate partnerships and projects.

Talk is Cheap

I've attended industry roundtables about how university programs are under-serving our profession. Although such roundtables are fun and spirited exchanges, we've been preaching to the proverbial choir. We must drive change and engage with university programs to make that change happen–all of us. Thus, I implore every one of you to do your part. Connect with your former or local schools and offer your experience and insights on the needs and directions of our industry. Offer to advise, present, share and establish partnerships and projects. Offer internships, and of course, hire their students. Engaging is how we pay it forward, and it's how we leave a legacy for the next generation to take what we have collectively developed and advance to the next level–and beyond.

Please visit our web site at www.infomanagementcenter.com for more information on these and other events.

Minimalism: Creating Information People Really Need
March 24 – April 28, 2022: Online Course
https://comtech-serv.com/training/minimalism/

How much redundant content do I really have? Actualizing the reuse in our documentation.
April 6, 2022: Webinar
https://www.infomanagementcenter.com/product/redundant-content/

The Value of Terminology in the Content Creation Process: How to define terminology and ensure its correct use in the authoring process
April 13, 2022 **:**Webinar
https://www.infomanagementcenter.com/product/Value-of-Terminology/

Why Your Whole Company Should Praise Your Tech Content – And How They Will
April 20, 2022: Webinar
https://www.infomanagementcenter.com/product/praise-your-content/

ConVEx Tempe
May 2 – 4, 2022: Tempe, Arizona
https://convex.infomanagementcenter.com/

Confab
May 9 – 12, 2022: Minneapolis, Minnesota
https://www.confabevents.com/

DITA Basics Training
May 12 – June 16, 2022: Online Course
https://comtech-serv.com/event/dita-basics-online-course/2022-05-12/

Information Modeling for Topic Based Authoring
May 12 – June 16, 2022: Online Course
https://comtech-serv.com/event/information-modeling-online_05-2022/

Advanced Reuse Strategies
May 12 – June 30, 2022: Online Course
https://comtech-serv.com/event/advanced-reuse-strategies/2022-05-12/

STC's Technical Communication Summit
May 15 – 18, 2022: Chicago, Illinois
https://summit.stc.org/conference/

Best Practices Conference
September 19 – 21, 2022: Baltimore, Maryland
https://bp.infomanagementcenter.com/

LavaCon
October 23 – 26, 2022: New Orleans, Louisiana
https://lavacon.org

acrolinx
Better Content. Faster.

Adobe
Technical Communication

BLUESTREAM
CONTENT SOLUTIONS

componize

DCL™
Data Conversion Laboratory Inc.

Etteplan

Fluid Topics

GlobalLink™
CCMS

Heretto

KALEIDOSCOPE
Taking your content global

IXIASOFT

precision content

Quanos
Content Solutions

RWS

stilo

[ZOOMIN]